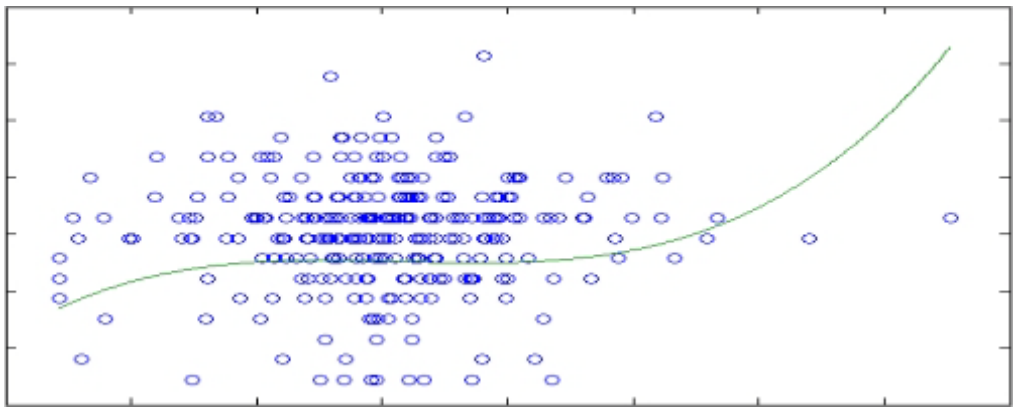
WarpPLS 5.0 User Manual



Ned Kock

WarpPLS© 5.0
User Manual

Ned Kock

ScriptWarp Systems™
Laredo, Texas
USA

WarpPLS 5.0 User Manual, January 2015, Copyright © by Ned Kock

All rights reserved worldwide. No part of this publication may be reproduced or utilized in any form, or by any means – electronic, mechanical, magnetic or otherwise – without permission in writing from ScriptWarp Systems.

Software use agreement

The use of the software that is the subject of this manual (Sofware) requires a valid license, which has a limited duration (usually no more than one year). Individual and organizational licenses may be purchased from ScriptWarp Systems, or any authorized ScriptWarp Systems reseller.

The Software is provided "as is", and without any warranty of any kind. Free trial versions of the Software are made available by ScriptWarp Systems with the goal of allowing users to assess, for a limited time (usually one to three months), the usefulness of the Software for their data modeling and analysis purposes. Users are strongly advised to take advantage of those free trial versions, and ensure that the Software meets their needs before purchasing a license.

Free trial versions of the Software are full implementations of the software, minus the licenses. That is, they are not demo versions. Nevertheless, they are provided for assessment purposes only, and not for "production" purposes, such as to analyze data and subsequently publish it as a consulting or research report. Users must purchase licenses of the Software before they use it for "production" purposes.

Multivariate statistical analysis software systems are inherently complex, sometimes yielding results that are biased and disconnected with the reality of the phenomena being modeled. Users are strongly cautioned against accepting the results provided by the Software without double-checking those results against: past empirical results obtained by other means and/or with other software, applicable theoretical models, and practical commonsense assumptions.

Under no circumstances is ScriptWarp Systems to be held liable for any damages caused by the use of the Software. ScriptWarp Systems does not guarantee in any way that the Software will meet the needs of its users.

For more information:

ScriptWarp Systems
P.O. Box 452428
Laredo, Texas, 78045
USA
www.scriptwarp.com

Table of contents

A. INTRODUCTION **5**

A.1. Software installation and uninstallation 6
A.2. Stable version notice 7
A.3. New features in version 5.0 8
A.4. Experimental features 10

B. THE MAIN WINDOW **11**

B.1. The SEM analysis steps 12
B.2. Data 14
B.3. Modify 16
B.4. Settings 18
B.5. Data labels 20
B.6. General settings 21
B.7. Data modification settings 28
B.8. Individual inner model algorithm settings 30
B.9. Individual latent variable weight and loading starting value settings 31
B.10. Grouped descriptive statistics 32

C. STEP 1: OPEN OR CREATE A PROJECT FILE TO SAVE YOUR WORK **34**

D. STEP 2: READ THE RAW DATA USED IN THE SEM ANALYSIS **36**

E. STEP 3: PRE-PROCESS THE DATA FOR THE SEM ANALYSIS **38**

F. STEP 4: DEFINE THE VARIABLES AND LINKS IN THE SEM MODEL **40**

F.1. Create or edit the SEM model 41
F.2. Create or edit latent variable 44

G. STEP 5: PERFORM THE SEM ANALYSIS AND VIEW THE RESULTS **46**

H. VIEW AND SAVE RESULTS **48**

H.1. View general results 50
H.2. View path coefficients and P values 54
H.3. View standard errors and effect sizes for path coefficients 56
H.4. View indicator loadings and cross-loadings 58
H.5. View indicator weights 62
H.6. View latent variable coefficients 65
H.7. View correlations among latent variables and errors 68
H.8. View block variance inflation factors 70
H.9. View correlations among indicators 72
H.10. View/plot linear and nonlinear relationships among latent variables 73
H.11. View indirect and total effects 80
H.12. View causality assessment coefficients 82

I. CONCLUDING REMARKS AND ADDITIONAL ISSUES **86**

I.1. Warping from a conceptual perspective 87
I.2. Interpreting warped relationships 89
I.3. Correlation versus collinearity 91
I.4. Stable P value calculation methods 93
I.5. Missing data imputation methods 95
I.6. Factor-Based PLS algorithms 97

J. GLOSSARY **99**

K. ACKNOWLEDGEMENTS **103**

L. REFERENCES **104**

A. Introduction

Structural equation modeling (SEM) employing the partial least squares (PLS) method, or PLS-based SEM for short, has been and continue being extensively used in a wide variety of fields. Examples of fields in which PLS-based SEM is used are information systems (Guo et al., 2011; Kock & Lynn, 2012), marketing (Biong & Ulvnes, 2011), international business (Ketkar et al., 2012), nursing (Kim et al., 2012), medicine (Berglund et al., 2012), and global environmental change (Brewer et al., 2012).

This software provides users with a wide range of features, several of which are not available from other SEM software. For example, this software is the first and only (at the time of this writing) to explicitly identify nonlinear functions connecting pairs of latent variables in SEM models and calculate multivariate coefficients of association accordingly.

Additionally, this software is the first and only (at the time of this writing) to provide classic PLS algorithms together with factor-based PLS algorithms for SEM (Kock, 2014). Factor-based PLS algorithms generate estimates of both true composites and factors, fully accounting for measurement error. They are equivalent to covariance-based SEM algorithms; but bring together the "best of both worlds", so to speak.

Factor-based PLS algorithms combine the precision of covariance-based SEM algorithms under common factor model assumptions (Kock, 2014) with the nonparametric characteristics of classic PLS algorithms. Moreover, factor-based PLS algorithms address head-on a problem that has been discussed since the 1920s – the factor indeterminacy problem. Classic PLS algorithms yield composites, as linear combinations of indicators, which can be seen as factor approximations. Factor-based PLS algorithms, on the other hand, provide estimates of the true factors, as linear combinations of indicators and measurement errors.

All of the features provided have been extensively tested with both "real" data, collected in actual empirical studies, as well as simulated data generated through Monte Carlo procedures (Robert & Casella, 2010). Future tests, however, may reveal new properties of these features, and clarify the nature of existing properties.

A.1. Software installation and uninstallation

The software installs automatically from a self-extracting executable file. There are two components to the software: the MATLAB Compiler Runtime, and the main software (i.e., WarpPLS). The first is a set of free-distribution MATLAB libraries with code that is called by the main software. Because the MATLAB Compiler Runtime is used, you do *not* have to have MATLAB (the main MATLAB program) installed on your computer to run WarpPLS.

Minimal and harmless changes to the operating system registry are made by the MATLAB Compiler Runtime, which are easily reversed upon uninstallation. To uninstall, go the "Control Panel", click on "Add or Remove Programs" or "Programs and Features", and uninstall the MATLAB Compiler Runtime.

The **MATLAB Compiler Runtime 7.14** is used in this version of WarpPLS. This is the same MATLAB Compiler Runtime as the one used in versions 2.0 – 4.0. The MATLAB Compiler Runtime used in version 1.0 is a different one, and thus will *not* work properly with this version of WarpPLS.

In most cases, previous versions of WarpPLS and of the MATLAB Compiler Runtime may be retained on a user's computer. Different versions of WarpPLS and of the MATLAB Compiler Runtime generally do not interfere with one other.

To uninstall the main software program, simply delete the main software installation folder. This folder is usually "C:\Program Files\WarpPLS 5.0" or "C:\Program Files (x86)\WarpPLS 5.0", unless you chose a different folder for the main software program during the installation process. Then delete the shortcut created by the software from the desktop.

Both programs, the MATLAB Compiler Runtime and the main software, may be retained without harm to your computer. They will not normally interfere with other programs; not even with MATLAB (the main MATLAB program), if you have it installed on your computer.

A.2. Stable version notice

This version was initially released as a beta version and was later upgraded to stable. As you will see below, **it incorporates a large number of new features**, when compared with the previous version. It has undergone extensive testing in-house prior to its release as a beta version, and has been in the hands of users for several months prior to its upgrade to stable. Nevertheless, **given the large number of new features, and the inherent interconnectedness of features, it is very likely that this stable version will still contain some software bugs, which we expect to be minor**.

A.3. New features in version 5.0

Factor-Based PLS algorithms. There has been a long and in some instances fairly antagonistic debate among proponents and detractors of the use of Wold's original PLS algorithms in the context of SEM. This debate has been fueled by one key issue: Wold's original PLS algorithms do not deal with actual factors, as covariance-based SEM algorithms do; but with composites, which are exact linear combinations of indicators (Kock, 2014d). The new "factor-based" algorithms provided in this version have been developed specifically to address this perceived limitation of Wold's original PLS algorithms. These new algorithms are called Factor-Based PLS Type CFM1, Factor-Based PLS Type REG1, and Factor-Based PLS Type PTH1.

New descriptive statistics for indicators and latent variables. An extended set of descriptive statistics is now provided for both indicators and latent variables. The descriptive statistics provided include: minimum and maximum values, medians, modes, skewness and excess kurtosis coefficients, as well as results of unimodality and normality tests. These are now complemented by histograms, which can be viewed on the screen and saved as files.

Unimodality and normality tests. Often the use of PLS-based SEM methods is justified based on them making no data normality assumptions, but typically without any accompanying test of normality! This is addressed in this version through various outputs of unimodality and normality tests, which are now provided for all indicators and latent variables. The unimodality tests performed are the Rohatgi- Székely test and the Klaassen-Mokveld-van Es test. The normality tests performed are the classic Jarque-Bera test and Gel & Gastwirth's robust modification of this test. Where these tests are applied to individual indicators, they can be seen as "univariate" or "bivariate" unimodality and normality tests. Where these tests are applied to latent variables, they can be seen as "multivariate" unimodality and normality tests.

New 3D graphs and graph rotation. Rocky and smooth 3D graphs can now be viewed with data points excluded. Corresponding graphs with data points included are also available. The 3D graph displays with data points excluded are analogous to those used in the focused 2D graphs. Additionally, users can now incrementally rotate 3D graphs in the following directions: up, down, left, and right.

New "stable" P value calculation methods. An extended set of "stable" P value calculation methods is now available to users: Stable1, Stable2, and Stable3. The Stable1 method was the software's default up until version 4.0, when it was called simply the "stable" method. The Stable2 and Stable3 methods have been developed as alternatives to the Stable1 method that rely on the direct application of exponential smoothing formulas (for the formulas, see: Kock, 2014b), and that can thus be more easily implemented and tested by methodological researchers. A Monte Carlo experiment shows that the Stable2 and Stable3 methods yield estimates of the actual standard errors that are consistent with those obtained via bootstrapping, in many cases yielding more precise estimates of the actual standard errors (Kock, 2014b). The more accurate of the two methods appears to be the Stable3 method, which is now the software's default.

New missing data imputation methods. Several missing data imputation methods are now available to users: Arithmetic Mean Imputation (the software's default), Multiple Regression Imputation, Hierarchical Regression Imputation, Stochastic Multiple Regression Imputation, and Stochastic Hierarchical Regression Imputation. The missing data imputation method chosen by the user will be employed in the execution of Step 3, and also after that when the option "Redo missing data imputation (via data pre-processing)" is selected. The option is available under the

"Modify" menu options. Kock (2014c) provides a detailed discussion of these methods, as well as of a Monte Carlo simulation whereby the methods' relative performances are investigated.

Incremental code optimization. This is conducted in each new version of this software. At several points the code was optimized for speed, stability, and coefficient estimation precision. This led to incremental gains in speed even as a number of new features were added. Several of these new features required new and complex calculations, mostly to generate coefficients that were not available before.

A.4. Experimental features

Some of the features provided by the software are still at an experimental stage, and may change in the future as more tests are conducted. Normally this is indicated in this user manual whenever it is the case.

Other novel features of this software may prove useful for applications different from the ones they were originally intended for. For example, **an extensive set of causality assessment coefficients is provided by the software. Yet, the topic of causality assessment in the context of SEM is controversial (Pearl, 2009).** A causality assessment coefficient that is provided to inform the user of the possibility of a reverse link may prove in the future to be useful to identify a specific type of bias due to measurement error.

Finally, while this software aims at providing a wide range of features and outputs, **the ubiquity of measurement error in SEM analyses (as well as data analyses in general)** would tend to make **strong and sweeping claims regarding accuracy and statistical power likely to be proven exaggerated or even wrong.**

Researchers analyzing empirical data typically **do not know the underlying distributions of their data and of error terms**. Data analysis **software tools help researchers uncover characteristics of those distributions, with incomplete information**. Given this, it seems reasonable to conclude that all SEM algorithms and software tools that implement these algorithms have limitations in their accuracy, avoidance of false positives, and statistical power (i.e., avoidance of false negatives).

Accuracy and statistical power seem to suffer particularly when very small samples and deviations from normality are observed in the context of small effect sizes. Goodhue et al.'s (2012) extensive analysis of various SEM algorithms illustrates these limitations, although its negative results may have been exacerbated by the fairly low path coefficients that they used for small and medium effect sizes. Those path coefficients were based on effect sizes that were calculated using the stepwise regression procedure proposed by Cohen (1988) for the calculation of *f*-squared coefficients, which is generally not compatible with PLS-based SEM algorithms. This theme is further explored later in this user manual.

This software attempts to ameliorate this situation in connection with accuracy and statistical power by providing an extensive set of features and outputs that can be used by researchers to reveal as many aspects of the underlying relationships as possible.

Some of the features provided are specifically aimed at increasing accuracy and statistical power. For example, Jackknifing, one of the resampling methods provided, tends to generate relatively low standard errors with small samples and medium to high effect sizes. This could increase statistical power with small samples and medium to high effect sizes, making the use of Jackknifing more appropriate than Bootstrapping in these cases. The same may be true for the "stable" methods, particularly Stable3. In fact, Monte Carlo simulations suggest that the "stable" methods perform better than Jackknifing in this respect.

This software's extensive range of features may also help further research on SEM methods in general.

B. The main window

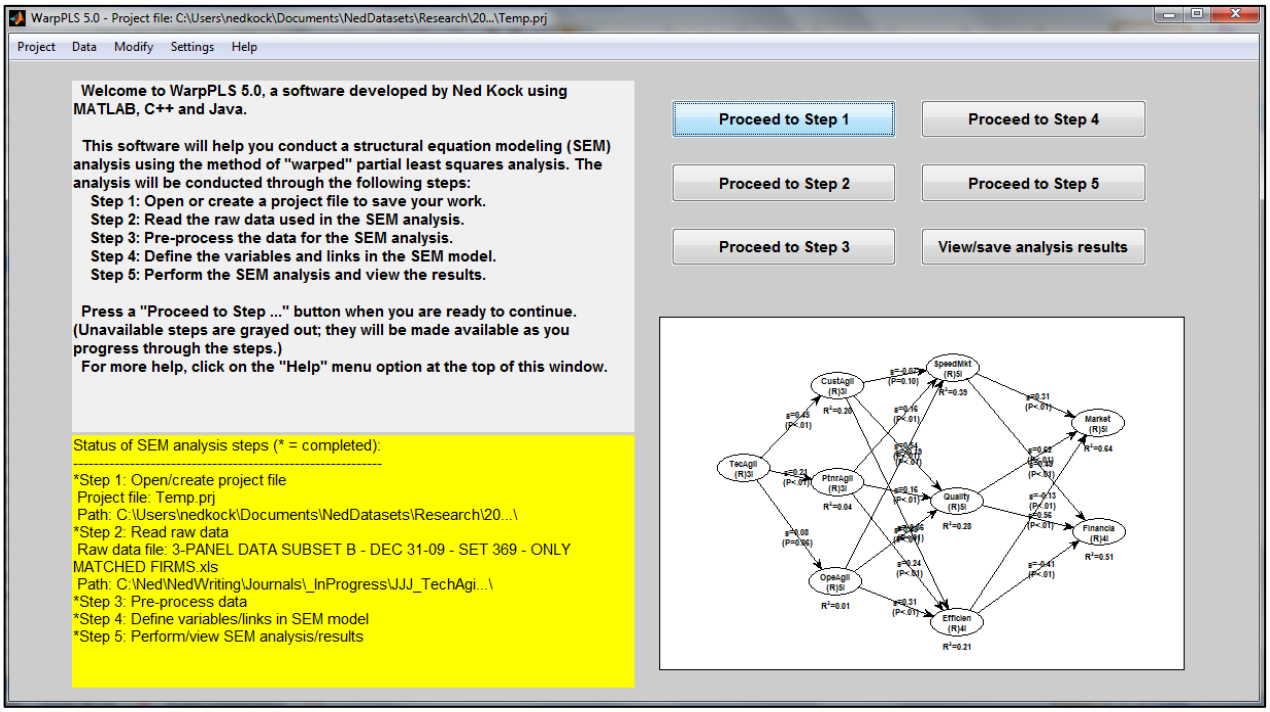
Prior to displaying the software's main window, **a command prompt window may be shown and kept open for the duration of the SEM analysis session**. Do not try to close this command prompt window, if it is shown, because it will list warnings and error messages that will likely be very useful in troubleshooting. Moreover, those warnings and error messages will indicate where in the source code they are happening, which will help the software developer correct any possible problems in future versions of the software.

In very slow computers, with limited computing power, only the command prompt window may be displayed for as long as a few minutes. The reason for this is that the computer needs to load a large runtime module prior to actually running this software. Users should not try to do anything during this time, as that will only delay the launch of the software's main window.

B.1. The SEM analysis steps

The software's main window (see Figure B.1.2) is where the SEM analysis starts. The top-left part of the main window contains a brief description of the five steps through which the SEM analysis takes place. The steps are executed by pressing each of the push buttons on the top-right part of the window. Not all menu options and push buttons become available right away. Menu options and push buttons become available as the analysis progresses.

Figure B.1.2. The main window showing the steps (after a complete analysis was conducted)


WarpPLS 5.0 - Project file: C:\Users\nedkock\Documents\NedDatasets\Research\20...\Temp.prj
Project
Data
Modify
Settings
Help
Welcome to WarpPLS 5.0, a software developed by Ned Kock using MATLAB, C++ and Java.
This software will help you conduct a structural equation modeling (SEM) analysis using the method of "warped" partial least squares analysis. The analysis will be conducted through the following steps:
Step 1: Open or create a project file to save your work.
Step 2: Read the raw data used in the SEM analysis.
Step 3: Pre-process the data for the SEM analysis.
Step 4: Define the variables and links in the SEM model.
Step 5: Perform the SEM analysis and view the results.
Press a "Proceed to Step ..." button when you are ready to continue. (Unavailable steps are grayed out; they will be made available as you progress through the steps.)
For more help, click on the "Help" menu option at the top of this window.
Proceed to Step 1
Proceed to Step 2
Proceed to Step 3
Proceed to Step 4
Proceed to Step 5
View/save analysis results
Status of SEM analysis steps (* = completed):
*Step 1: Open/create project file
Project file: Temp.prj
Path: C:\Users\nedkock\Documents\NedDatasets\Research\20...\
*Step 2: Read raw data
Raw data file: 3-PANEL DATA SUBSET B - DEC 31-09 - SET 369 - ONLY MATCHED FIRMS.xls
Path: C:\Ned\NedWriting\Journals_InProgress\JJJ_TechAgi...\
*Step 3: Pre-process data
*Step 4: Define variables/links in SEM model
*Step 5: Perform/view SEM analysis/results

The steps must be carried out in the proper sequence. For example, Step 5, which is to perform the SEM analysis and view the results, cannot be carried out before Step 1 takes place, which is to open or create a project file to save your work. This is the main reason why steps have their push buttons grayed out and deactivated until it is time for the corresponding steps to be carried out.

The bottom-left part of the main window shows the status of the SEM analysis; after each step in the SEM analysis is completed, this status window is updated. A miniature version of the SEM model graph is shown at the bottom-right part of the main window. This miniature version is displayed without results after Step 4 is completed. After Step 5 is completed, this miniature version is displayed with results.

The "Project" menu options. There are three project menu options available: "Save project", "Save project as …", and "Exit". Through the **"Save project"** option you can save the project file that has just been created or that has been created before and is currently open. To open an existing project or create a new project you need to execute Step 1, by pressing the "Proceed to Step 1" push button. The **"Save project as …"** option allows you to save an existing project with a different name and in a different folder from ones for the project that is currently open or has just been created. This option is useful in the SEM analysis of multiple models where each

model is a small variation of the previous one. Finally, the **"Exit"** option ends the software session. If your project has not been saved, and you choose the "Exit" option, the software will ask you if you want to save your project before exiting. In some cases, you will not want to save your project before exiting, which is why a project is not automatically saved after each step is completed. For example, you may want to open an existing project, change a few things and then run a SEM analysis, and then discard that project. You can do this by simply not saving the project before exiting.

After Step 3 is completed, whereby the data used in the SEM analysis is pre-processed, three sets of menu options become available from the main window: **"Data"**, **"Modify"**, and **"Settings"**.

The "Data" menu options. These menu options allow you to view or save data and various statistics, mostly descriptive statistics, into tab-delimited .txt files. The "tab-delimited .txt file" is the general file format used by the software to save most of the files containing analysis and summarization results. These files can be opened and edited using Excel, Notepad, and other similar spreadsheet or text editing software. These menu options are discussed in more detail later.

The "Modify" menu options. These menu options allow you to add new data labels and raw data to your dataset, redo missing data imputation, as well as add one or more latent variable scores (a.k.a. factor scores) to the dataset as new standardized indicators. Also available is the option of adding all latent variable scores at once to the dataset as new standardized indicators. Data labels can be shown on graphs as text next to data points, or as legends for data points using different markers. These menu options are discussed in more detail later.

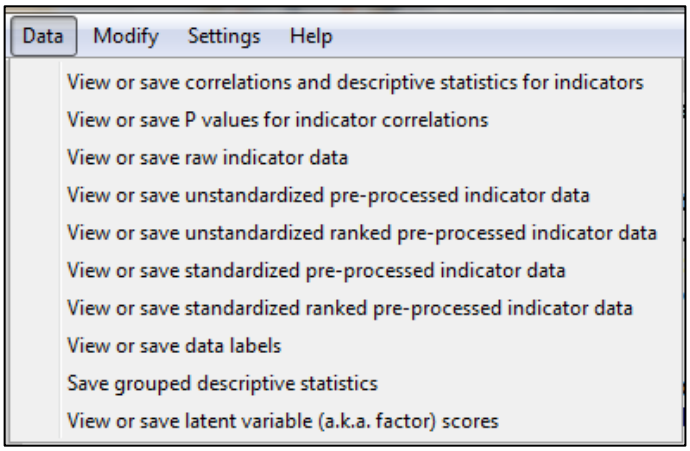
The "Settings" menu options. You can view or change general SEM analysis settings through the "Settings" menu options. Here you can select the analysis algorithms used in the SEM analysis, the resampling method used to calculate standard errors and P values, as well as other elements that will define how the SEM analysis will be conducted. These menu options are discussed in more detail later.

The "Help" menu options. There are several help menu options available on the main window, as well as on several other windows displayed by the software. The **"Open context-sensitive User Manual file (PDF)"** option opens this document as a PDF file from a Web location in a context-sensitive manner, in this case at an area that is specific to the main window. The **"Open User Manual file (PDF)"** option opens this document as a PDF file from a Web location, and is not context-specific. The **"Open Web page with video for this window"** option opens a Web page with a video clip that is context-specific, in this case specific to the main window. The **"Open Web page with links to various videos"** option is not context-specific, and opens a Web page with links to various video clips. The **"Open Web page with WarpPLS blog"** option opens a Web page with the WarpPLS blog. Similar help options are available from several other windows in this software.

B.2. Data

The "**Data**" menu options allow you to view or save data and various statistics, mostly descriptive statistics (see Figure B.2). These menu options are discussed individually below. Some of them are discussed in more detail later in this document.

Figure B.2. Data menu options


Data
Modify
Settings
Help
View or save correlations and descriptive statistics for indicators
View or save P values for indicator correlations
View or save raw indicator data
View or save unstandardized pre-processed indicator data
View or save unstandardized ranked pre-processed indicator data
View or save standardized pre-processed indicator data
View or save standardized ranked pre-processed indicator data
View or save data labels
Save grouped descriptive statistics
View or save latent variable (a.k.a. factor) scores

The **"View or save correlations and descriptive statistics for indicators"** option allows you to view or save general descriptive statistics about the data. These include the following, which are shown at the bottom of the table that is displayed through this option: means, standard deviations, minimum and maximum values, medians, modes, skewness and excess kurtosis coefficients, results of unimodality and normality tests, and histograms. The unimodality tests for which results are provided are the Rohatgi- Székely test (Rohatgi & Székely, 1989) and the Klaassen-Mokveld-van Es test (Klaassen et al., 2000). The normality tests for which results are provided are the classic Jarque-Bera test (Jarque & Bera, 1980; Bera & Jarque, 1981) and Gel & Gastwirth's (2008) robust modification of this test. Since these tests are applied to individual indicators, they can be seen as "univariate" or "bivariate" unimodality and normality tests.

These descriptive statistics are complemented by the option **"View or save P values for indicator correlations"**. This option may be useful in the identification of candidate indicators for latent variables through the anchor variable procedure discussed by Kock & Verville (2012). This can be done prior to defining the variables and links in a model. This can also be done after the model is defined and an analysis is conducted, particularly in cases where the results suggest outer model misspecification. Examples of outer model misspecification are instances in which indicators are mistakenly included in the model by being assigned to certain latent variables, and instances in which indicators are assigned to the wrong latent variables (Kock & Lynn, 2012; Kock & Verville, 2012).

The "**View of save raw indicator data**" option allows you to view or save the raw data used in the analysis. This is a useful feature for geographically distributed researchers conducting collaborative analyses. With it, those researchers do not have to share the raw data as a separate file, as that data is already part of the project file.

Two menu options allow you to view or save unstandardized pre-processed indicator data. This pre-processed data is not the same as the raw data, as it has already been through the automated missing value correction procedure in Step 3. The options that allow you to view or

save unstandardized pre-processed indicator data are: **"View or save unstandardized pre-processed indicator data"** and **"View or save unstandardized ranked pre-processed indicator data"**. The latter option refers to ranked data.

When data is ranked, typically the value distances that typify outliers in data on ratio scales, whether standardized or unstandardized, are significantly reduced. This effectively eliminates outliers from the data, without any decrease in sample size. Often some information is lost due to ranking – e.g., the distances among data points based on answers on ratio scales.

Two related menu options allow you to view or save standardized pre-processed indicator data: **"View or save standardized pre-processed indicator data"** and **"View or save standardized ranked pre-processed indicator data"**. The latter option ranks the data prior to standardizing it. Ranking often has little effect on ordinal data (e.g., data on Likert-type scales), and a major impact on ratio data (e.g., yearly income).

The options that refer to unstandardized data allow you to view or save pre-processed data prior to standardization. The options that refer to standardized data allow you to view or save pre-processed data after standardization; that is, after all indicators have been transformed in such a way that they have a mean of zero and a standard deviation of one.

The "**View or save data labels**" option allows you to view or save data labels. These are text identifiers that are entered by you separately, through one of the "Modify" menu options. Like the original numeric dataset, the data labels are stored in a table. Each column of this table refers to one data label, and each row to the corresponding row of the original numeric dataset. Data labels can later be shown on graphs, either next to each data point that they refer to, or as part of a graph's legend.

The **"Save grouped descriptive statistics"** option is a special option that allows you to save descriptive statistics (means and standard deviations) organized by groups defined based on certain parameters; this option is discussed in more detail at the end of this section.

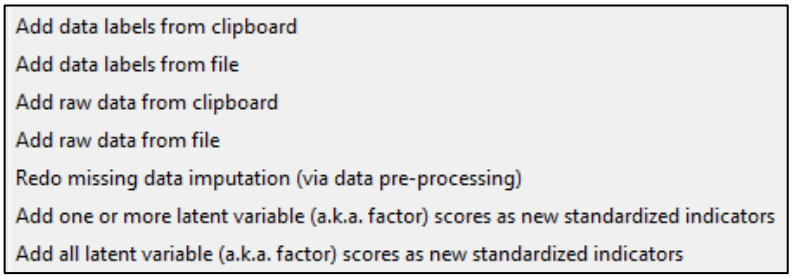
The "**View or save latent variable (a.k.a. factor) scores**" option allows you to view or save the latent variable scores generated by the software. There is another option that allows you to save latent variable scores, available as a menu option on the window used to view and save model analysis results; which becomes available later, after Step 5 is completed. These two options return the same latent variable scores in most cases.

The exception to the general rule above is a situation in which you specified a range restriction for your analysis. In this case, only the latter option will return the range-restricted latent variable scores. These latent variable scores will generally have a smaller number of rows than the original dataset (because they are range-restricted), and thus will not be exactly matched to the original dataset.

B.3. Modify

The "**Modify**" menu options allow you to add new data labels and raw data to your dataset, redo missing data imputation, as well as add one or more latent variable scores (a.k.a. factor scores) to the dataset as new standardized indicators (see Figure B.3). Also available is the option of adding all latent variable scores at once to the dataset as new standardized indicators. Data labels can be shown on graphs as text next to data points, or as legends for data points using different markers. These menu options are discussed individually below. Some of them are discussed in more detail later in this document.

Figure B.3. Modify menu options


Add data labels from clipboard
Add data labels from file
Add raw data from clipboard
Add raw data from file
Redo missing data imputation (via data pre-processing)
Add one or more latent variable (a.k.a. factor) scores as new standardized indicators
Add all latent variable (a.k.a. factor) scores as new standardized indicators

The menu options "**Add data labels from clipboard**" and "**Add data labels from file**" allow you to add data labels into the project file. Data labels are text identifiers that are entered by you through these options, one column at a time. Like the original numeric dataset, the data labels are stored in a table. Each column of this table refers to one data label, and each row to the corresponding row of the original numeric dataset. Data labels can later be shown on graphs, either next to each data point that they refer to, or as part of the legend for a graph.

Data labels can be read from the clipboard or from a file, but **only one column of labels can be read at a time**. Data label cells cannot be empty, contain spaces, or contain only numbers; **they must be combinations of letters, or of letters and numbers**. Valid examples are the following: "Age>17", "Y2001", "AFR", and "HighSuccess". These would normally be entered without the quotation marks, which are used here only for clarity. Some invalid examples: "123", "Age > 17", and "Y 2001".

Through the menu options "**Add raw data from clipboard**" and "**Add raw data from file**" users can add new data from the clipboard or from a file. This data then becomes available for use in models, without users having to go back to Step 2. These options relieve users from having to go through nearly all of the steps of a SEM analysis if they find out that they need more data after they complete Step 5 of the analysis. Past experience supporting users suggests that this is a common occurrence. These options employ the same data checks and data correction algorithms as in Step 2; please refer to the section describing that step for more details.

The option "**Redo missing data imputation (via data pre-processing)**" allows users to redo the missing data imputation process after choosing a method through the "**View or change missing data imputation settings**" option, which is available under the "**Settings**" menu options. The following missing data imputation methods are available: Arithmetic Mean Imputation (the software's default), Multiple Regression Imputation, Hierarchical Regression Imputation, Stochastic Multiple Regression Imputation, and Stochastic Hierarchical Regression

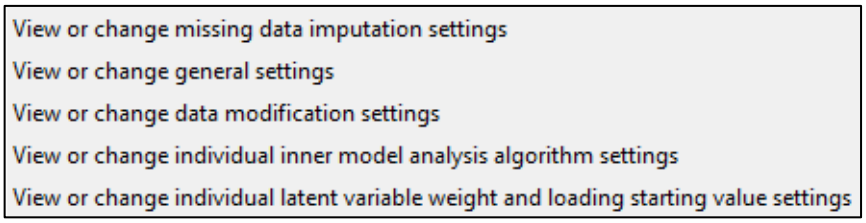
Imputation. Kock (2014c) provides a detailed discussion of these methods and of a Monte Carlo simulation that assesses the methods' relative performances.

Latent variable scores can be easily added to the dataset via the options "**Add one or more latent variable (a.k.a. factor) scores as new standardized indicators**" and "**Add all latent variable (a.k.a. factor) scores as new standardized indicators**". These options allow users, after Step 5 is completed, to add one or more latent variables to the model as new standardized indicators, and also to add all latent variables as new indicators. Adding one or more latent variables at a time may be advisable in certain cases; for example, in hierarchical analyses using selected latent variables as indicators of second, third etc. order latent variables at each level. In such cases, adding all latent variables at once may soon clutter the set of indicators available to be used in the SEM model.

B.4. Settings

The "**Settings**" menu options allow you to view or change general SEM analysis settings (see Figure B.4). Here you can select the analysis algorithm used in the SEM analysis, the resampling method used to calculate standard errors and P values, as well as other elements that will define how the SEM analysis will be conducted. These menu options are discussed individually below. Several of them are discussed in more detail later in this document.

Figure B.4. Settings menu options


View or change missing data imputation settings
View or change general settings
View or change data modification settings
View or change individual inner model analysis algorithm settings
View or change individual latent variable weight and loading starting value settings

The "**View or change missing data imputation settings**" option allows you to set the missing data imputation method to be used by the software, from among the following methods: Arithmetic Mean Imputation (the software's default), Multiple Regression Imputation, Hierarchical Regression Imputation, Stochastic Multiple Regression Imputation, and Stochastic Hierarchical Regression Imputation. The missing data imputation method chosen will be used **prior to execution of Step 3**, and also after that when the option "**Redo missing data imputation (via data pre-processing)**" under the "**Modify**" menu option is selected. Kock (2014c) provides a detailed discussion of these methods, as well as of a Monte Carlo simulation whereby the methods' relative performances are investigated.

The "**View or change general settings**" option allows you to set the outer model analysis algorithm, default inner model analysis algorithm, resampling method, and number of resamples. Through these sub-options, users can set outer and default inner model algorithms separately. Users are also allowed to set inner model algorithms for individual paths through a different option. If users choose not to set inner model algorithms for individual paths in an analysis of a new model (i.e., a model that has just been created), their choice of default inner model algorithm is automatically used for all paths.

The "**View or change data modification settings**" option allows you to select a range restriction variable type, range restriction variable, range (min-max values) for the restriction variable, and whether to use only ranked data in the analysis. Through these sub-options, users can run their analyses with subsamples defined by a range restriction variable, which is chosen from among the indicators available. They can also conduct their analyses with only ranked data, whereby all of the data is automatically ranked prior to the SEM analysis. When data on a ratio scale is ranked, typically the value distances that typify outliers are significantly reduced, effectively eliminating outliers without any decrease in sample size.

The "**View or change individual inner model analysis algorithm settings**" option allows you to set inner model algorithms for individual paths. That is, for each path a user can select a different algorithm from among the following choices: "Linear", "Warp2", "Warp2 Basic", "Warp3", and "Warp3 Basic".

The "**View or change individual latent variable weight and loading starting value settings**" option allows you to set the initial values of the weights and loadings for each latent

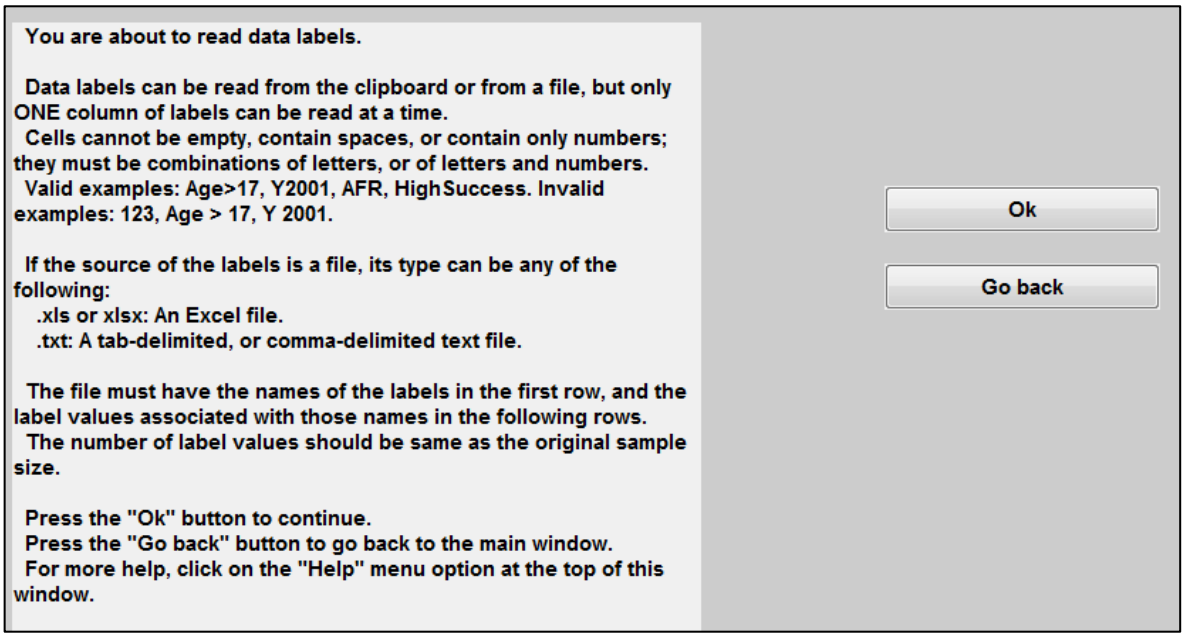
variable. The default is 1 for all weights and loadings. With this option, latent variables measured in a reversed way, as well as formative latent variables with most of their weights and loadings ending up being negative, can be more easily operationalized.

Several of the options above, and their component elements, are discussed in more detail in the subsections below, still in this section describing the main window options. These subsections include further discussions about data labels, general settings, data modification settings, individual inner model analysis algorithm settings, as well as individual latent variable weight and loading starting value settings. A further discussion of grouped descriptive statistics, which can be saved through a sub-option under the “Data” options, is also provided.

B.5. Data labels

Data labels can be added through the menu options "**Add data labels from clipboard**" and "**Add data labels from file**" (see Figure B.5). Data labels are text identifiers that are entered by you through these options, one column at a time. Like the original numeric dataset, the data labels are stored in a table. Each column of this table refers to one data label, and each row to the corresponding row of the original numeric dataset. Data labels can later be shown on graphs, either next to each data point that they refer to, or as part of the legend for a graph. Once they have been added, data labels can be viewed or saved using the "**View or save data labels**" option.

Figure B.5. Add data labels from file window

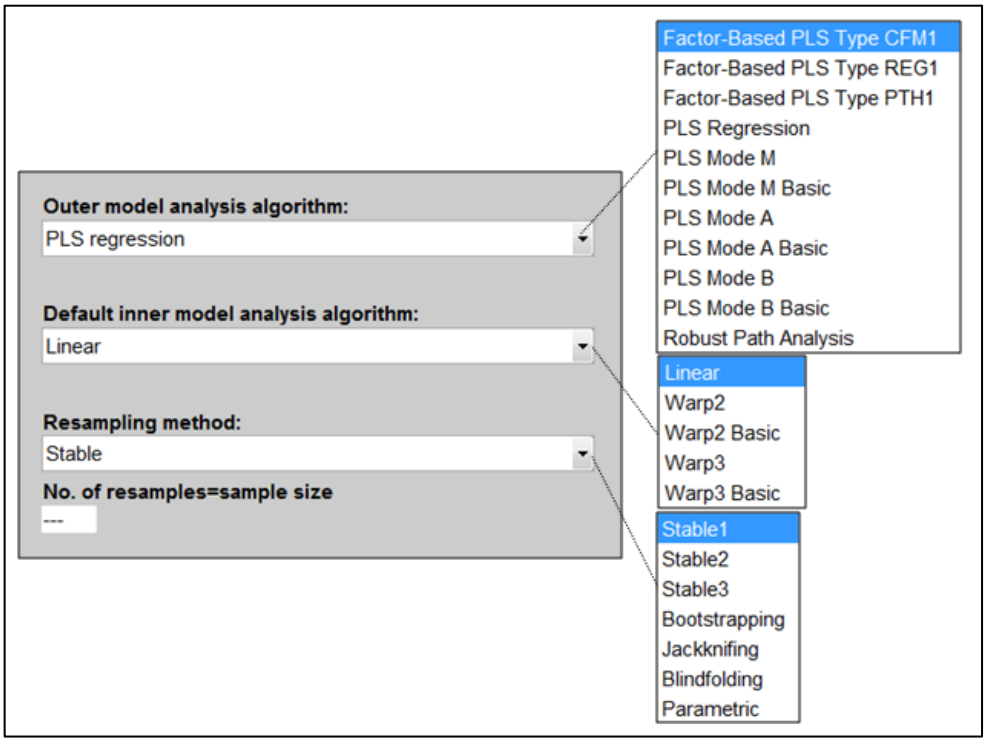

You are about to read data labels.
Data labels can be read from the clipboard or from a file, but only ONE column of labels can be read at a time.
Cells cannot be empty, contain spaces, or contain only numbers; they must be combinations of letters, or of letters and numbers.
Valid examples: Age>17, Y2001, AFR, HighSuccess. Invalid examples: 123, Age > 17, Y 2001.
If the source of the labels is a file, its type can be any of the following:
.xls or xlsx: An Excel file.
.txt: A tab-delimited, or comma-delimited text file.
The file must have the names of the labels in the first row, and the label values associated with those names in the following rows.
The number of label values should be same as the original sample size.
Press the "Ok" button to continue.
Press the "Go back" button to go back to the main window.
For more help, click on the "Help" menu option at the top of this window.
Ok
Go back

While data labels can be read from the clipboard or from a file, **only one column of labels can be read at a time**. Data label cells cannot be empty, contain spaces, or contain only numbers; **they must be combinations of letters, or of letters and numbers**. Valid examples are the following: "Age>17", "Y2001", "AFR", and "HighSuccess". These would normally be entered without the quotation marks, which are used here only for clarity. Some invalid examples are: "123", "Age > 17", and "Y 2001".

B.6. General settings

The "**View or change general settings**" option allows users to set the outer model analysis algorithm, default inner model analysis algorithm, resampling method, and number of resamples (see Figure B.6). Through these sub-options, users can set outer and default inner model algorithms separately. Users are also allowed to set inner model algorithms for individual paths, but through a different settings option. If users choose not to set inner model algorithms for individual paths, their choice of default inner model algorithm is automatically used for all paths.

Figure B.6. View or change general settings window



The settings chosen for each of the options can have a dramatic effect on the results of a SEM analysis. At the same time, the right combinations of settings can provide major insights into the data being analyzed. As such, the settings' options should be used with caution, and normally after a new project file (with a unique name) is created and the previous one saved. This allows users to compare results and, if necessary, revert back to project files with previously selected settings.

A key criterion for the calculation of the weights, observed in virtually all classic PLS-based algorithms, is that the regression equation expressing the relationship between the indicators and the latent variable scores has an error term that equals zero. In other words, in classic PLS-based algorithms the latent variable scores are calculated as exact linear combinations of their indicators. This is not the case with the new Factor-Based PLS algorithms provided by this software, as these new algorithms estimate latent variable scores fully accounting for measurement error.

The warping takes place during the estimation of path coefficients, and after the estimation of all weights, latent variable scores, and loadings in the model. The weights and loadings of a

model with latent variables make up what is often referred to as the **outer model (a.k.a. measurement model)**, whereas the path coefficients among latent variables make up what is often called the **inner model (a.k.a. structural model)**.

The **outer model analysis algorithms** available are **Factor-Based PLS Type CFM1, Factor-Based PLS Type REG1, Factor-Based PLS Type PTH1, PLS Regression, PLS Mode M, PLS Mode M Basic, PLS Mode A, PLS Mode A Basic, PLS Mode B, PLS Mode B Basic, and Robust Path Analysis**. All of these outer model algorithms share a common characteristic. They calculate latent variable scores as exact linear combinations of their indicators, or of their indicators and measurement errors. With the exception of the Robust Path Analysis algorithm, all of these algorithms perform iterations until they converge to a solution.

There has been a long and in some instances fairly antagonistic debate among proponents and detractors of the use of Wold's original PLS algorithms (Adelman & Lohmoller, 1994; Lohmöller, 1989; Wold, 1980) in the context of SEM. This debate has been fueled by one key issue, which is analogous to the issue underlying the related principal components versus factor analysis debate. **Wold's original PLS algorithms do not deal with actual factors**, as covariance-based SEM algorithms do; **but with composites**, which are exact linear combinations of indicators (Kock, 2014d). **The Factor-Based PLS algorithms** provided by this software **have been developed specifically to address this perceived limitation** of Wold's original PLS algorithms.

Factor-Based PLS Type CFM1 generates estimates of both true composites and factors, in two stages, explicitly accounting for measurement error (Kock, 2014). Like covariance-based SEM algorithms, **this algorithm is fully compatible with common factor model assumptions**, including the assumption that all indicator errors are uncorrelated. In its first stage, this algorithm employs a new "true composite" estimation sub-algorithm, which estimates composites based on mathematical equations that follow directly from the common factor model. The second stage employs a new "variation sharing" sub-algorithm, which can be seen as a "soft" version of the classic expectation-maximization algorithm (Dempster et al., 1977) used in maximum likelihood estimation, with apparently faster convergence and nonparametric properties.

Factor-Based PLS Type REG1 and **Factor-Based PLS Type PTH1** are also factor-based PLS algorithms that generate estimates of both composites and factors, in two stages, fully accounting for measurement error. The **Factor-Based PLS Type REG1** algorithm first estimates composites via PLS Regression (discussed below), and then estimates factors employing variation sharing (Kock, 2014). Among the factor-based algorithms available in this software, this Factor-Based PLS Type REG1 algorithm can be seen as the closest to Wold's original PLS design.

The **Factor-Based PLS Type PTH1** algorithm first estimates composites via Robust Path Analysis (discussed below), and then estimates factors employing variation sharing (Kock, 2014). By doing so, this algorithm addresses several of the concerns about Wold's original PLS algorithms raised in an important critical article by Rönkkö & Evermann (2013). This algorithm can also be seen as addressing the call for simplicity in a thought-provoking article on PLS by Rigdon (2012). Unlike the Factor-Based PLS Type CFM1algorithm, the **Factor-Based PLS Type REG1** and **Factor-Based PLS Type PTH1** algorithms **do not impose certain common factor model assumptions** that rarely hold in practice, such as the assumption that all indicator errors are uncorrelated.

PLS Regression has been the default outer model algorithm since the software's inception, and is maintained as such as a matter of tradition. This algorithm iterates until the outer model

weights become stable with the following calculations being performed in successive iterations for each latent variable in the model: (a) the outer model weights are calculated through a least squares regression where the latent variable is the predictor and the indicators are the criteria; and (b) the latent variable is calculated as an exact linear combination of the indicator scores. In the PLS Regression algorithm, **the inner model does not influence the outer model**. That is, the weights are not influenced by the links connecting latent variables, which are created by the user in Step 4.

The following outer model algorithms are similar to PLS Regression, but in them **the inner model influences the outer model**: **PLS Mode M, PLS Mode M Basic, PLS Mode A, PLS Mode A Basic, PLS Mode B, and PLS Mode B Basic**. These are classic PLS algorithms that have been historically associated with PLS-based SEM software (Chatelin et al., 2002; Temme et al., 2006). In them, the iterative process leading to the calculation of latent variable scores involves the intermediate calculation of path coefficients, correlations, and signs of correlations. These are used as inputs in the calculation of weights in successive iterations, typically leading to the addition of collinearity among latent variables that are linked.

The above collinearity inflation that occurs when the inner model influences the outer model often has the effect of strengthening associations among linked latent variables. This is a real phenomenon that has been presented as a weakness of PLS-based SEM, and that has been referred to as the "capitalization on error" problem of PLS-based algorithms (see, e.g., Goodhue et al., 2012). This problem is generally overstated, as PLS-based algorithms in general tend to also reduce collinearity. That is, when the inner model influences the outer model collinearity is indeed increased, but often not to the extent that the increase offsets the previous collinearity decrease that normally results from the use of PLS-based algorithms.

PLS Mode M is often referred as the "MIMIC" or "mixed" mode. In it, the inner model influences the outer model through path coefficients and correlations, depending on whether the links go into or out from each latent variable, respectively. PLS Mode M in fact uses either PLS Mode A or PLS Mode B, based on whether latent variables are defined as reflective or formative, respectively. The PLS modes A and B are discussed below.

PLS Mode M Basic is a variation of PLS Mode M in which the inner model influences the outer model through the signs of correlations among latent variables. This corresponds to what Lohmöller (1989) refers to as a "basic scheme", also referred to as a "centroid scheme" (Tenenhaus et al., 2005).

For the purposes of PLS-based SEM, the **schemes known as "centroid" and "factorial" are largely redundant** (Tenenhaus et al., 2005), but they share a common property. **They tend to reduce the number of instances of Simpson's paradox** (Wagner, 1982) in the SEM analysis results. Because of this property and the fact that these two schemes are redundant, this software implements only one of them, the "centroid" scheme. This scheme is referred to as "basic", for simplicity and consistency with prior seminal publications that set the foundations of PLS-based SEM (see, e.g., Lohmöller, 1989).

PLS Mode A is often referred to as the "reflective" mode, which is arguably incorrect because both reflective and formative latent variables can be used with this algorithm. In other words, using PLS Mode A does not make a formative latent variable become a reflective latent variable. In it, the inner model influences the outer model through path coefficients and correlations, depending on whether the links go into or out from each latent variable, respectively. In this mode the outer model weights are calculated through a least squares regression where the latent variable is the predictor and the indicators are the criteria. **PLS Mode A Basic** is a variation of

PLS Mode A in which the inner model influences the outer model through the signs of the correlations among latent variables.

PLS Mode B is often referred to as the "formative" mode. This is arguably incorrect for the same reason discussed above, namely that both reflective and formative latent variables can be used with this algorithm. In other words, using PLS Mode B does not turn a reflective latent variable into a formative latent variable. However, PLS Mode B is often less stable than PLS Mode A, and also tends to cause a significant increase in collinearity among linked latent variables. In it, the inner model influences the outer model through path coefficients and correlations, depending on whether the links go into or out from each latent variable, respectively. In this mode the outer model weights are calculated through a least squares regression where the indicators are the predictors and the latent variable the criterion. **PLS Mode B Basic** is a variation of PLS Mode B in which the inner model influences the outer model through the signs of the correlations among latent variables.

The **Robust Path Analysis** algorithm is a simplified algorithm in which latent variable scores are calculated by averaging the scores of the indicators associated with the latent variables. That is, in this algorithm weights are not estimated through PLS Regression. This algorithm is called "robust" path analysis, because the P values can be calculated through the nonparametric resampling or stable methods implemented through the software. If all latent variables are measured with single indicators, the Robust Path Analysis algorithm will yield latent variable scores and outer model weights that are identical to those generated through the other algorithms.

Many relationships in nature, including relationships involving behavioral variables, are nonlinear and follow a pattern known as U-curve (or inverted U-curve). In this pattern a variable affects another in a way that leads to a maximum or minimum value, where the effect is either maximized or minimized, respectively. This type of relationship is also referred to as a J-curve pattern; a term that is more commonly used in economics and the health sciences.

The term **"U-curve" is used here also to refer to nonlinear relationships that can be represented as sections of a U curve**. As such, it covers all noncyclical nonlinear relationships. These relationships **include the logarithmic, hyperbolic decay, exponential decay, exponential, and quadratic relationships**, among others. That is, these relationships can be conceptually modeled as variations of U-curve relationships.

The **default inner model analysis algorithms** available are the following: **Linear, Warp2, Warp2 Basic, Warp3, and Warp3 Basic**. All of these inner model algorithms share a common characteristic. They calculate path coefficients through least squares regression algorithms based on the latent variable scores calculated through one of the outer model analysis algorithms available.

The **Linear** algorithm does not perform any warping of relationships. The **Warp2** algorithm tries to identify U-curve relationships among linked latent variables, and, if those relationships exist, the algorithm transforms (or "warps") the scores of the predictor latent variables so as to better reflect the U-curve relationships in the estimated path coefficients in the model. Here the signs of the path coefficients are initially (i.e. prior to the inner model least squares regressions) assigned as the signs of the corresponding path coefficients obtained without any warping. Similarly to the outer model "basic" versions, the **Warp2 Basic** algorithm is a variation of the Warp2 algorithm that tends to reduce the number of instances of Simpson's paradox (Wagner, 1982) in the final results. This happens because in this basic version the signs of path coefficients are initially assigned as the signs of the corresponding correlations obtained without any warping.

The **Warp3** algorithm, the default algorithm used by the software, tries to identify relationships among latent variables defined by functions whose first derivatives are U-curves. These types of relationships follow a pattern that is more similar to an S-curve (or a somewhat distorted S-curve). An S-curve can be seen as a combination of two connected U-curves, one of which is inverted. Examples of S-curve functions are the sigmoid, hyperbolic sine and hyperbolic tangent. The logistic function is a type of sigmoid function, and thus is also an example of S-curve function. Similarly to the Warp2 Basic algorithm, the **Warp3 Basic** algorithm is a variation of the Warp3 algorithm that tends to reduce the number of instances of Simpson's paradox (Wagner, 1982) in the final results. Again here this happens because the signs of path coefficients are initially assigned as the signs of the corresponding correlations obtained without any warping.

In summary, with the exception of the Linear algorithm, all of the default inner model analysis algorithms perform nonlinear transformations on the predictor latent variable scores prior to the calculation of path coefficients. In other words, except for the Linear algorithm, these algorithms "warp" the predictor latent variable scores by finding best-fitting nonlinear functions that minimize sums of squared residuals on a bivariate basis. This process can be seen as another least squares minimization stage that is "in between" those used in the calculation of latent variable scores and path coefficients.

One of several resampling methods may be selected for the calculation of P values and related coefficients (e.g., standard errors). In the calculation of P values, a one-tailed test is generally recommended if the coefficient is assumed to have a sign (positive or negative), which should be reflected in the hypothesis that refers to the corresponding association (Kock, 2014d). Hence this software reports one-tailed P values for coefficients used in hypothesis testing (e.g., path coefficients); from which two-tailed P values can be easily obtained if needed (Kock, 2014d). The available **resampling methods** are the following: **Stable1, Stable2, and Stable3, Bootstrapping, Jackknifing, Blindfolding, and Parametric**.

With the **Stable1** method, the software's default up until version 4.0 (when it was called simply the "stable" method), P values are calculated through nonlinear fitting of standard errors to empirical standard errors generated with the other resampling methods available. In other words, the Stable1 method could be viewed as a quasi-parametric method that yields P values that try to approximate the "average" P values generated by the software's other resampling methods.

The **Stable2** and **Stable3** methods have been developed as alternatives to the Stable1 method. Unlike the Stable1 method, they rely on the direct application of exponential smoothing formulas (for details, see: Kock, 2014b), and that can thus be more easily implemented and tested by methodological researchers. A Monte Carlo experiment shows that the **Stable2** and **Stable3** methods yield estimates of the actual standard errors that are consistent with those obtained via bootstrapping, in many cases yielding more precise estimates of the actual standard errors (Kock, 2014b). The more accurate of the two methods appears to be the Stable3 method, which also appears to be more accurate than the Stable1 method. As such, **the Stable3 method is set as the software's default** starting in version 5.0.

With the **Parametric** method, P values are calculated assuming multivariate normality and also that path coefficient estimates are distributed as expected based on the central limit theorem. Neither the **Parametric** method nor the three "stable" methods (**Stable1**, **Stable2** and **Stable3**) actually generates resamples, so calling them resampling methods is done here for simplicity in the grouping of settings options. Because no resamples are generated, these are the most efficient

of the methods from a computing load perspective. These methods can be particularly useful in the analysis of large datasets, as in these cases creating resamples can be computationally very taxing. With the emergence of the concept of "big data", the need to analyze large datasets is becoming increasingly common.

Bootstrapping employs a resampling algorithm that creates a number of resamples (a number that can be selected by the user), by a method known as "resampling with replacement". This means that each resample contains a random arrangement of the rows of the original dataset, where some rows may be repeated. The commonly used analogy of a deck of cards being reshuffled, leading to many resample decks, is a good one; but not entirely correct because in Bootstrapping the same "card" may appear more than once in each of the resample "decks".

Jackknifing, on the other hand, creates a number of resamples that equals the original sample size, and where each resample has one row removed. That is, the sample size of each resample is the original sample size minus 1. Thus, **when Jackknifing is selected the number of resamples is automatically set as the sample size**. This refers to the most common form of jackknifing, also known as "delete-1" and "classic" jackknifing, which is the one implemented through this software.

Blindfolding employs a resampling algorithm that creates a number of resamples (a number that can be selected by the user) by a method whereby each resample has a certain number of rows replaced with the means of the respective columns. The number of rows modified in this way in each resample equals the sample size divided by the number of resamples. For example, if the sample size is 200 and the number of resamples selected is 100, then each resample will have 2 rows modified. If a user chooses a number of resamples that is greater than the sample size, the number of resamples is automatically set to the sample size (as with Jackknifing).

The default number of resamples for Bootstrapping and Blindfolding is 100. It can be modified by entering a different number in the appropriate edit box. (Please note that we are talking about the number of resamples here, not the sample size of the original dataset.) **Leaving the number of resamples for Bootstrapping as 100 is recommended** because it has been shown that higher numbers of resamples lead to negligible improvements in the reliability of P values (see, e.g., Goodhue et al., 2012). In fact, according to the original developer of the Bootstrapping method, even setting the number of resamples at 50 is likely to lead to fairly reliable P value estimates (Efron et al., 2004).

Conversely, increasing the number of resamples well beyond 100 leads to a higher computation load on the software, making the software look like it is having a hard time coming up with the results. In very complex models, a high number of resamples may make the software run very slowly. Some researchers have suggested in the past that a large number of resamples can address problems with the data, such as the presence of outliers due to errors in data collection. This opinion is not shared by the original developer of the Bootstrapping method, Bradley Efron (see, e.g., Efron et al., 2004).

Not considering the "stable" methods, arguably Jackknifing is particularly good at addressing problems associated with the presence of outliers due to errors in data collection. Generally speaking, Jackknifing tends to generate more stable resample path coefficients (and thus more reliable P values) with small sample sizes (lower than 100), and with samples containing outliers (see, e.g., Chiquoine & Hjalmarsson, 2009). Monte Carlo simulations suggest that the "stable" methods perform better than Jackknifing in this respect.

Again, not considering the "stable" methods, Bootstrapping tends to generate more stable resample path coefficients (and thus more reliable P values) with larger samples and with

samples where the data points are evenly distributed on a scatter plot. Monte Carlo simulations suggest that the "stable" methods perform better than Bootstrapping in this respect as well. The use of Bootstrapping with small sample sizes (lower than 100) has been discouraged (Nevitt & Hancock, 2001).

Generally speaking, Bootstrapping and Jackknifing can be seen as complementary resampling methods, in that one tends to perform well in situations where the other does not, and vice-versa. Nevertheless, the "stable" methods provided by this software seem to be an improvement over them, as indicated by Monte Carlo simulations (Kock, 2014b).

Blindfolding tends to perform somewhere in between Jackknifing and Bootstrapping. If the number of resamples is set as very close to the sample size, particularly with small sample sizes (lower than 100) and with samples containing outliers, Blindfolding performs similarly to Jackknifing. With larger samples and with samples where the data points are evenly distributed on a scatter plot, Blindfolding tends to performs more like Bootstrapping, especially when the number of resamples is set as the same for both algorithms.

Prior to the development of the "stable" methods, a recommendation was usually made in connection with Bootstrapping and Jackknifing. Since the warping algorithms are also sensitive to the presence of outliers, the recommendation was to estimate P values with both Bootstrapping and Jackknifing, which are complementary resampling methods, and use the P values associated with the most stable coefficients. An indication of instability is a high P value (i.e., statistically non-significant) associated with path coefficients that could be reasonably expected to yield low P values. For example, with a sample size of 100, a path coefficient of 0.2 could be reasonably expected to yield a P value that is statistically significant at the 0.05 level. If that is not the case, there may be a stability problem. Another indication of instability is a marked difference between the P values estimated through Bootstrapping and Jackknifing.

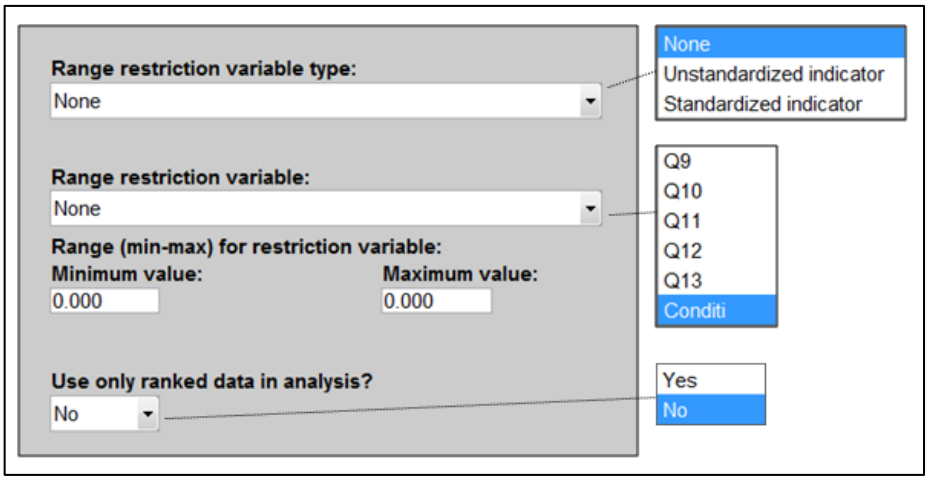
The recommendation above was based on the fact that P values can be easily estimated using two or more resampling methods by following the simple procedure outlined as follows. Run a SEM analysis of the desired model, using one of the resampling methods, and save the project. Then save the project again, this time with a different name, change the resampling method, and run the SEM analysis again. Then save the second project again. Each project file will now have results that refer to one of the resampling methods. The P values can then be compared, and the most stable ones used in a research report on the SEM analysis. While this is a perfectly valid approach for the calculation of P values, as the coefficients to which the P values refer do not change across iterations, it is very important to fully disclose this to the readers of the research report (or reports) written based on the SEM analyses.

An alternative to the above approach is the use one of the "stable" methods, particularly the **Stable3** method (see, e.g., Kock, 2014b), as these methods can be seen as yielding P values that are consistent with and often more precise than the P values generated by the software's other resampling methods. Using these "stable" methods has the advantage of requiring much less manual work from the user. Based on various tests in the context of PLS-based SEM, it seems that the Stable3 method yields fairly reliable results for path coefficients associated with direct effects (Kock, 2014b). It is less clear if the Stable3 method, or any of the other "stable" methods, is advisable for the calculation of P values for path coefficients associated with indirect and total effects, and research in this area is ongoing.

B.7. Data modification settings

The "**View or change data modification settings**" option allows users to run their analyses with subsamples defined by a **range restriction variable**, which is chosen from among the indicators available. (After Step 5 is completed, latent variable scores can also be added to the model as standardized indicators.). This option also allows users to conduct their analyses with only **ranked data** (see Figure B.7).

Figure B.7. View or change data modification settings



Two **range restriction variable types** are available: standardized and unstandardized indicators. This means that the range restriction variable can be either a standardized or unstandardized indicator. **Once a range restriction variable is selected, minimum and maximum values must be set (i.e., a range)**, which in turn has the effect of restricting the analysis to the rows in the dataset within that particular range.

The option of selecting a range restriction variable and respective range is useful in multi-group analyses, whereby separate analyses are conducted for group-specific subsamples, saved as different project files, and the results then compared against one another. One example would be a multi-country analysis, with each country being treated as a subsample, but without separate datasets for each country having to be provided as inputs.

Let us assume that an unstandardized variable called "Country" stores the values "1" (for Brazil), "2" (for New Zealand), and "3" (for the USA). To run the analysis only with data from Brazil one can set the range restriction variable as "Country" (after setting its type as "Unstandardized indicator"), and then set both the minimum and maximum values as "1" for the range.

This range restriction feature is also useful in situations where outliers are causing instability in a resample set, which can lead to abnormally high standard errors and thus inflated P values. Users can remove outliers by restricting the values assumed by a variable to a range that excludes the outliers, without having to modify and re-read a dataset.

Users can also select an option to conduct their analyses with only **ranked data**, whereby all of the data is automatically ranked prior to the SEM analysis (the original data is retained in unranked format). When data measured on ratio scales is ranked, typically the value distances that typify outliers are significantly reduced, effectively eliminating outliers without any

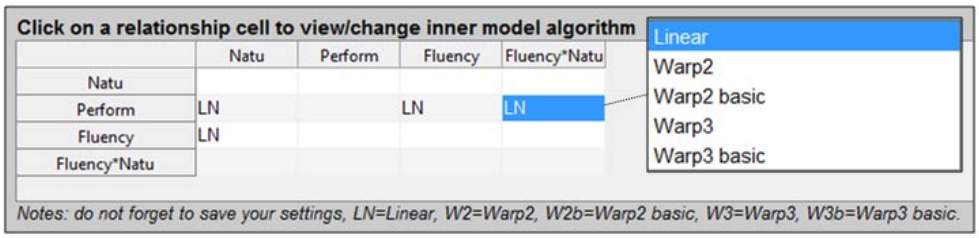
decrease in sample size. Contrary to popular belief, this cannot be achieved through standardization only.

Often some information is lost due to ranking – e.g., the distances among data points based on answers on ratio scales. Thus a concomitant increase in collinearity may be observed, but typically not to the point of threatening the credibility of the results. The option of using only ranked data in the analysis can be very useful in assessments of whether the presence of outliers significantly affects path coefficients and respective P values, especially when outliers are not believed to be due to measurement error.

B.8. Individual inner model algorithm settings

The "**View or change individual inner model algorithm settings**" option allows users to set inner model algorithms for individual paths (see Figure B.8). The algorithms available are the same as those that can be selected as default inner model analysis algorithms**: Linear, Warp2, Warp2 Basic, Warp3, and Warp3 Basic**.

Figure B.8. View or change individual inner model algorithm settings



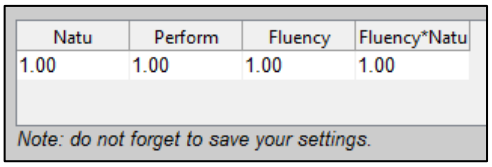
Individual inner model algorithms can be set for both regular and interaction effect latent variables; the latter are associated with moderating effects. If no choice is made for an individual inner model algorithm, the default inner model analysis algorithm is used. If a model is changed after an analysis is conducted, the individual inner model algorithms are set to the default inner model analysis algorithm.

This option allows users to customize their analyses based on theory and past empirical research. If theory or results from past empirical research suggest that a specific link between two latent variables is linear, then the corresponding path can be set to be analyzed using the Linear algorithm. Conversely, if theory or results from past empirical research suggest that a specific link between two latent variables should have the shape of a U curve (or J curve), the corresponding path can be set to be analyzed using the Warp2 algorithm or the Warp2 Basic algorithm.

B.9. Individual latent variable weight and loading starting value settings

The "**View or change individual latent variable weight and loading starting value settings**" option allows users to set the initial values of the weights and loadings for each latent variable (see Figure B.9). This is a specialized option that will only rarely be used. The default starting value for all latent variables is 1. While any real number can be used here, normally only -1 and 1 are used.

Figure B.9. View or change individual latent variable weight and loading starting value settings



This option reflects a little known characteristic of classic PLS-based SEM analyses, which is that they do not always converge to the same solution. The estimated coefficients depend on the starting values of weights and loadings, thus leading to different solutions depending on the initial configurations of those starting values. Even in simple models, often at least two solutions exist – as long as latent variables are used, with multiple indicators. By convention the solution most often accepted as valid is the one associated with the default starting value for all latent variables, which is 1.

With this option, latent variables measured in a reversed way can be more easily operationalized. An example would be a latent variable reflecting boredom being measured through a set of indicators that individually reflect excitement. In this type of scenario, generally the starting value of weights and loadings for the latent variable should be set to -1.

This option can also be useful with formative latent variables for which most of the weights and loadings end up being negative after an analysis is conducted. In this case, paths associated with the latent variable may end up being reversed, leading to conclusions that are the opposite of what is hypothesized. The solution here would normally be a change in sign for starting value of weights and loadings, usually from 1 to -1.

B.10. Grouped descriptive statistics

When the **"Save grouped descriptive statistics"** option is selected, a data entry window is displayed (see Figure B.10.1). There you can choose a grouping variable, number of groups, and the variables to be grouped. This option is useful if one wants to conduct a comparison of means analysis using the software, where one variable (the grouping variable) is the predictor, and one or more variables are the criteria (the variables to be grouped).

Figure B.10.1. Save grouped descriptive statistics window

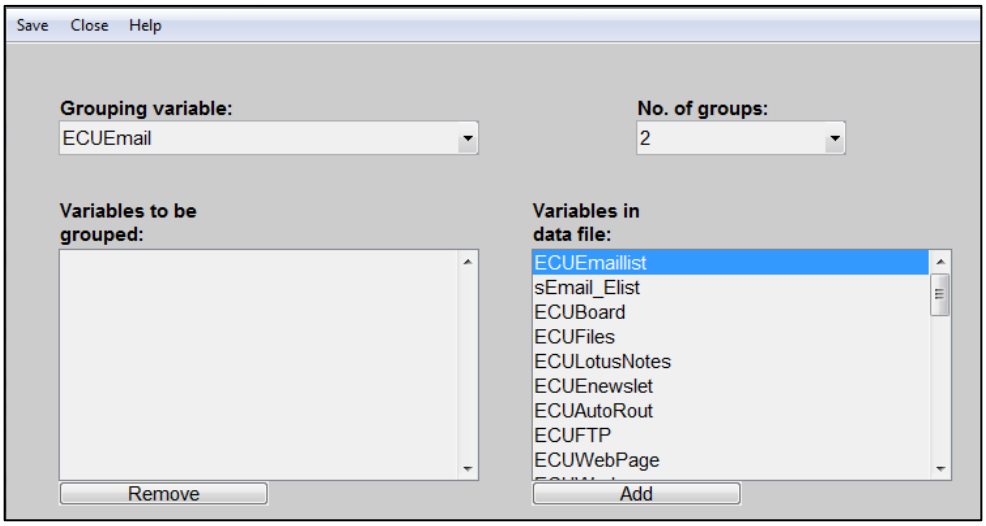


Figure B.10.2. Grouped descriptive statistics bar chart

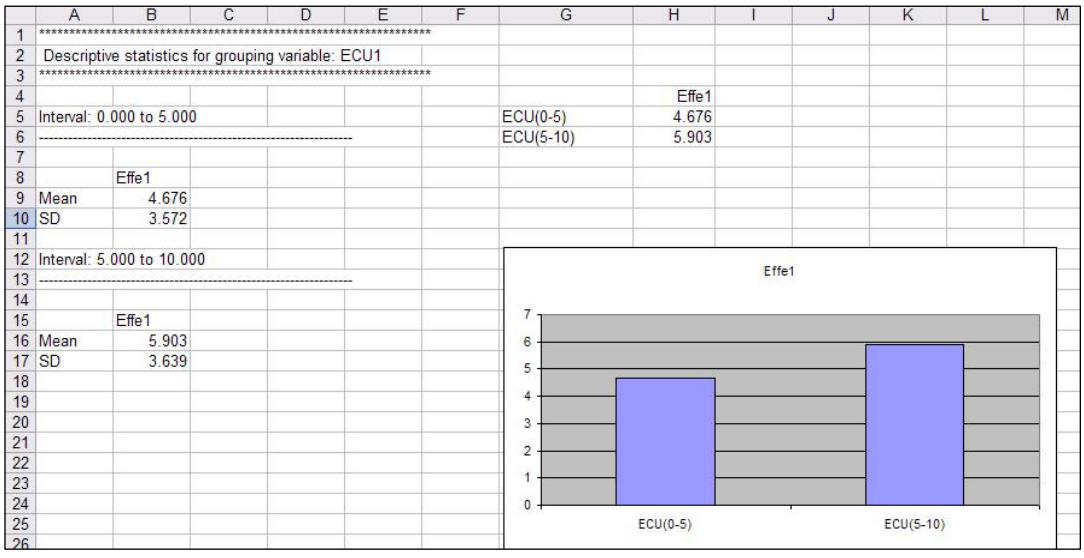


Figure B.10.2 shows the grouped statistics data saved through the window shown in Figure B.10.1. The tab-delimited .txt file was opened with a spreadsheet program, and contained the data on the left part of the figure.

The data on the left part of Figure B.10.2 was organized as shown above the bar chart; next the bar chart was created using the spreadsheet program's charting feature. If a simple comparison of

means analysis using this software had been conducted in which the grouping variable (in this case, an indicator called “ECU1”) was the predictor, and the criterion was the indicator called “Effe1”, those two variables would have been connected through a path in a simple path model with only one path. Assuming that the path coefficient was statistically significant, the bar chart displayed in Figure B.10.2, or a similar bar chart, could be added to a report describing the analysis.

Some may think that it is overkill to conduct a comparison of means analysis using a SEM software package such as this, but there are advantages in doing so. One of those advantages is that this software calculates P values using a nonparametric class of estimation techniques, namely resampling and “stable” estimation techniques. (Resampling techniques are sometimes referred to as bootstrapping techniques, which may lead to confusion since bootstrapping is also the name of a type of resampling technique.) Nonparametric estimation techniques do not require the data to be normally distributed, which is a requirement of other comparison of means techniques (e.g., ANOVA).

Another advantage of conducting a comparison of means analysis using this software is that the analysis can be significantly more elaborate than with traditional comparison of means methods, even nonparametric ones. For example, the analysis may include control variables (or covariates), which would make it equivalent to an ANCOVA test. Finally, the comparison of means analysis may include latent variables, as either predictors or criteria. This is not usually possible with ANOVA or commonly used nonparametric comparison of means tests (e.g., the Mann-Whitney U test).

An even more extreme situation is that discussed by Kock (2013b) where data on only “one group and one condition” is available. This situation is illustrated through a scenario in which a researcher obtains empirical data by asking questions to gauge the effect of a technology on task performance, but does not obtain data on the extent to which the technology is used. Because of this, the researcher ends up with only one column of data to analyze.

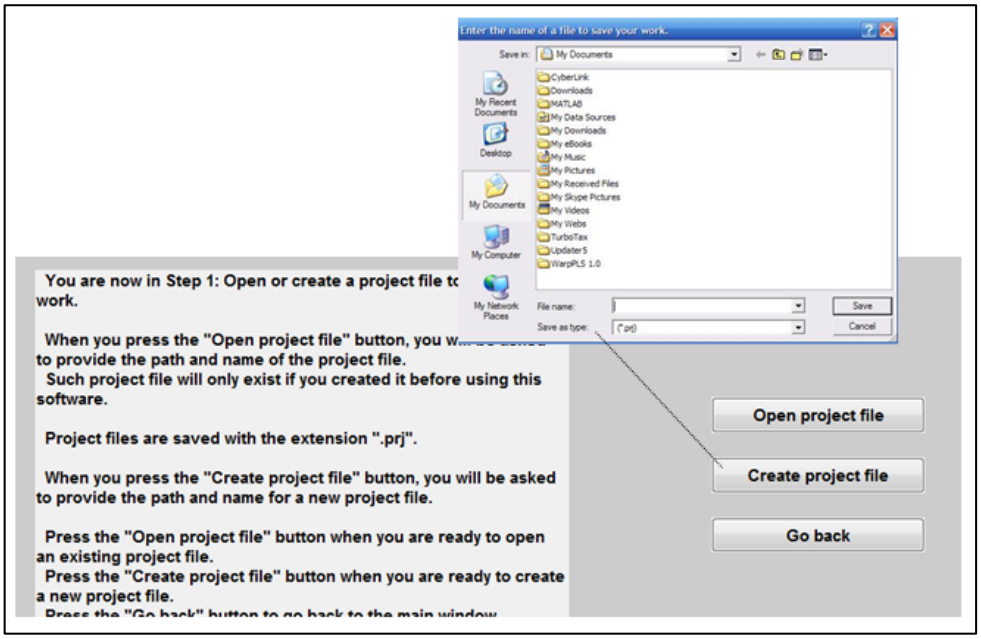
Two other scenarios are also discussed by Kock (2013b). These two scenarios are discussed to set the stage for the discussion of the “one group and one condition” scenario. The first is a typical study scenario in which the researcher measures the degree to which the technology is used, or the degree to which specific features of the technology are used, as well as team performance and/or related variables expected to be influenced by technology use.

In the second scenario the researcher does not have data on the extent to which the technology is used, but has data related to team performance and/or other variables expected to be influenced by technology use before and after the technology is introduced. This is a longitudinal data collection scenario for which a comparison of means test could be used. Data analyses for all three scenarios are discussed by Kock (2013b) based on this software, showing the versatility of the software. The main reason for this versatility is that most of the data analysis methods used in behavioral research can be conceptually seen as special cases of SEM.

C. Step 1: Open or create a project file to save your work

In Step 1 you will open or create a project file to save your work (see Figure C.1). **Project files are saved with the ".prj" extension, and contain all of the elements needed to perform a SEM analysis**. That is, they contain the original data used in the analysis, the graphical model, the inner and outer model structures, and the results.

Figure C.1. Step 1 window


Enter the name of a file to save your work.
Save in:
My Documents
File name:
Save as type:
Save
Cancel
You are now in Step 1: Open or create a project file to
work.
When you press the "Open project file" button, you w
to provide the path and name of the project file.
Such project file will only exist if you created it before using this
software.
Project files are saved with the extension ".prj".
When you press the "Create project file" button, you will be asked
to provide the path and name for a new project file.
Press the "Open project file" button when you are ready to open
an existing project file.
Press the "Create project file" button when you are ready to create
a new project file.
Open project file
Create project file
Go back

Once an original data file is read into a project file, the original data file can be deleted without effect on the project file. The project file will store the original location and file name of the data file so that this information is available in case it is needed in the future, but the project file will no longer use the data file.

Project files may be created with one name, and then renamed using Windows Explorer or another file management tool. Upon reading a project file that has been renamed in this fashion, the software will detect that the original name is different from the file name, and will adjust accordingly the name of the project file that it stores internally.

Different users of this software can easily exchange project files electronically if they are collaborating on a SEM analysis project. This way they will have access to all of the original data, intermediate data, and SEM analysis results in one single file. Project files are relatively small. For example, a complete project file of a model containing 5 latent variables, 32 indicators (columns in the original dataset), and 300 cases (rows in the original dataset) will typically be only approximately 200 KB in size. Simpler models may be stored in project files as small as 50 KB.

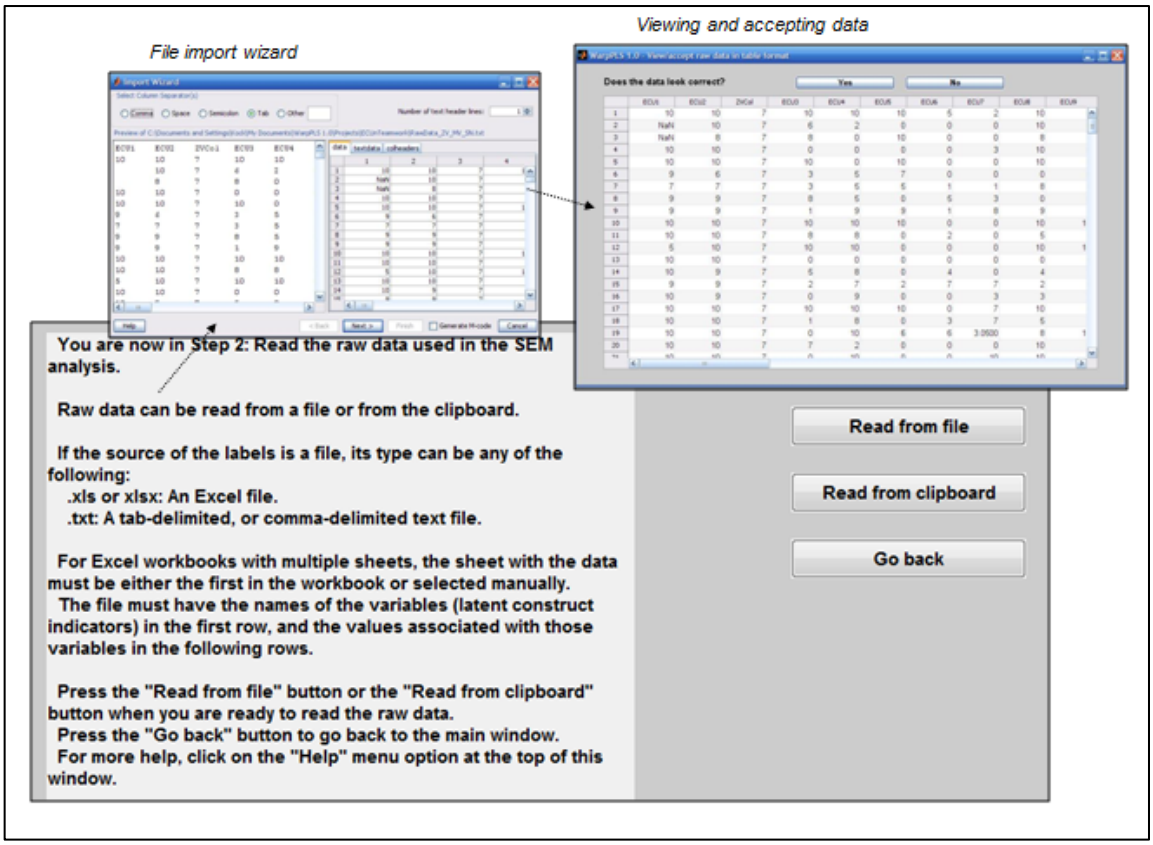
If a project file created with a previous version of the software is open, the software automatically recognizes that and converts the file to the new version. This takes placed even with project files where all of the five steps of the SEM analysis were completed. However, because each new version incorporates new features, with outputs stored within new or modified

software objects, normally **previous versions of the software cannot properly reuse project files created with more recent versions**.

D. Step 2: Read the raw data used in the SEM analysis

Through Step 2, you will read the raw data used in the SEM analysis (see Figure D.1). While this should be a relatively trivial step, it is in fact one of the steps where users have the most problems with other SEM software. Often a SEM software application will abort, or freeze, if the raw data is not in the exact format required by the SEM software, or if there are any problems with the data, such as missing values (empty cells).

Figure D.1. Reading the raw data used in the SEM analysis


File import wizard
Viewing and accepting data
You are now in Step 2: Read the raw data used in the SEM analysis.
Raw data can be read from a file or from the clipboard.
If the source of the labels is a file, its type can be any of the following:
.xls or xlsx: An Excel file.
.txt: A tab-delimited, or comma-delimited text file.
For Excel workbooks with multiple sheets, the sheet with the data must be either the first in the workbook or selected manually.
The file must have the names of the variables (latent construct indicators) in the first row, and the values associated with those variables in the following rows.
Press the "Read from file" button or the "Read from clipboard" button when you are ready to read the raw data.
Press the "Go back" button to go back to the main window.
For more help, click on the "Help" menu option at the top of this window.
Read from file
Read from clipboard
Go back

The buttons "**Read from file**" and "**Read from clipboard**" allow you to read raw data into the project file from a file or from the clipboard, respectively. This software employs an import wizard that avoids most data reading problems, even if it does not entirely eliminate the possibility that a problem will occur. **Click only on the "Next" and "Finish" buttons of the file import wizard, and let the wizard do the rest**. Soon after the raw data is imported, it will be shown on the screen, and you will be given the opportunity to accept or reject it. If there are problems with the data, such as missing column names, simply click "No" when asked if the data looks correct.

Raw data can be read directly from Excel files, with extensions ".xls" or ".xlsx", or text files where the data is tab-delimited or comma-delimited. **When reading from an ".xls" or ".xlsx" file that contains a workbook with multiple worksheets, make sure that the worksheet that contains the data is the first on the workbook**. If the workbook has multiple worksheets, the file import wizard used in Step 2 will typically select the first worksheet as the source or raw

data. If the desired worksheet is not the first in the workbook, in many cases the user will be able to select the proper worksheet through the wizard, but this selection can lead to mistakes when made by novice users. **Raw data files, whether Excel or text files, must have indicator names in the first row, and numeric data in the following rows**. They may contain empty cells, or missing values; these will be automatically replaced with values calculated by one of the missing data imputation algorithms available in a later step.

The "**View or change missing data imputation settings**" option under "**Settings**" allows you to set the missing data imputation method to be used by the software in the next step. Users may want to employ non-automated approaches to deal with missing data, such as deleting the rows with missing cells, or manually replacing them with the average of nearby values on the same column. The most widely used approach, and also a reasonably reliable one in the context of PLS-based SEM, is replacing the missing values with column averages. This missing data imputation method is called Arithmetic Mean Imputation, and is automated by the software. It is in fact the software's default missing data imputation method. Kock (2014c) provides a detailed discussion of various missing data imputation methods, as well as of a Monte Carlo simulation whereby the methods' relative performances are investigated.

While missing data imputation is done automatically by the software, you should not use datasets with too many missing values, as this will distort the results. A general rule of thumb is that your dataset should not have any column with more than 10 percent of its values missing; a more relaxed rule would be to set the threshold to 20 percent (Hair et al., 1987; 2009). On the other hand, **Kock (2014c) shows that even 30 percent of missing data will still not lead to significant bias (from the perspective of theory testing) with any of the missing data imputation methods employed by this software**. One can reduce the percentage of missing values per column by deleting rows in the dataset, where the deleted rows are the ones that refer to the columns with missing values.

One simple test can be used to try to find out if there are problems with a raw data file. Try to open it with a spreadsheet software (e.g., Excel), if it is originally a text file; or try to create a tab-delimited text file with it, if it is originally a spreadsheet file. If you try to do either of these things, and the data looks corrupted (e.g., missing column names, with misplaced columns, containing unrecognized symbols), then it is likely that the original file has problems, which may be hidden from view. For example, a spreadsheet file may be corrupted, but that may not be evident based on a simple visual inspection of the contents of the file.

E. Step 3: Pre-process the data for the SEM analysis

In Step 3 the raw data will be pre-processed for the SEM analysis. This is mostly an automatic process, requiring only a few button clicks from you. This step will correct problems with the data, such as: identical column names, columns with zero variance, and missing data.

The "View or change missing data imputation settings" option allows you to set the missing data imputation method to be used by the software in this step. Missing data imputation can be redone after this step, if you later decide to use a different imputation method. To accomplish that you should use the option "Redo missing data imputation (via data pre-processing)" under the "Modify" menu option. Kock (2014c) provides a detailed discussion of missing data imputation methods, as well as of a Monte Carlo simulation comparing the methods' relative performances.

This step will also let you know if the data has rank problems, which usually happens when the sample size is very small relative to the number of existing indicators. A related cause of rank problems is a sample with many repeated or linearly dependent values on different rows or columns, which sometimes is an indication of data fabrication. Please note that the term "rank" here comes from matrix algebra, and is unrelated to the same term used in the context of ranked data, as discussed earlier in connection with the software settings.

If there are rank problems, this does not mean that you cannot proceed with the SEM analysis. However, the results may be unstable and, in some cases, completely unreliable. On the other hand, it is not uncommon for rank problems to be reported and still the results of the ensuing SEM analysis turn out to be reliable. This is due to the general robustness of PLS-based methods for SEM analysis.

At the end of this step, a window will be displayed with the pre-processed data, which will be standardized. **Standardized data columns have means that equal zero and standard deviations that equal one.** If you use the Arithmetic Mean Imputation method for dealing with missing data (the software's default), previously missing values will be shown as zero, since they were replaced with the averages (or means) of the columns. **Standardized data usually ranges from -4 to 4, with outliers assuming values toward the left or right end of those extremes, sometimes beyond -4 or 4.**

Outliers can significantly change the shape of a nonlinear relationship, but this may also be the case with linear relationships. For example, one single outlier may change the sign of a linear association, from positive to negative (i.e., changing the relationship from direct to inverse). Because of this, there is invariably the temptation of removing outliers from analyses.

This is often a mistake (Giaquinta, 2009; Hair et al., 2009), as outliers can be invaluable in elucidating the true nature of an association (Kaiser, 2010; Rosenthal & Rosnow, 1991; Wold et al., 2001). Generally speaking, outliers should only be removed if there are good reasons to believe that they are due to measurement error.

After the software displays the pre-processed and standardized data, typically you will accept the data and move on to the next step. If the data looks corrupted, do not accept it; click on the "No" button when asked if the data looks correct. If there are problems in this step, they will usually be related to problems with the raw data file. Check that file, and see if you can correct those problems.

As mentioned before in this manual, one simple test can be used to try to find out if there are problems with a raw data file. Try to open it with a spreadsheet program, if it is originally a text file; or to try to create a tab-delimited text file with it, if it is originally a spreadsheet file. If you

try to do either of these things, and the data looks “messed up” (e.g., corrupted, or missing column names), then it is likely that the original file has problems, which may be hidden from view. For example, a spreadsheet file may be corrupted, but that may not be evident based on a simple visual inspection of the contents of the file using spreadsheet software.

F. Step 4: Define the variables and links in the SEM model

In Step 4 you will define the latent variables and links in the SEM model. The sub-steps that make up this step are discussed in more detail in the subsections below. This software employs a graphical interface that allows users to create and edit model elements visually and directly; i.e., without the need of a scripting language.

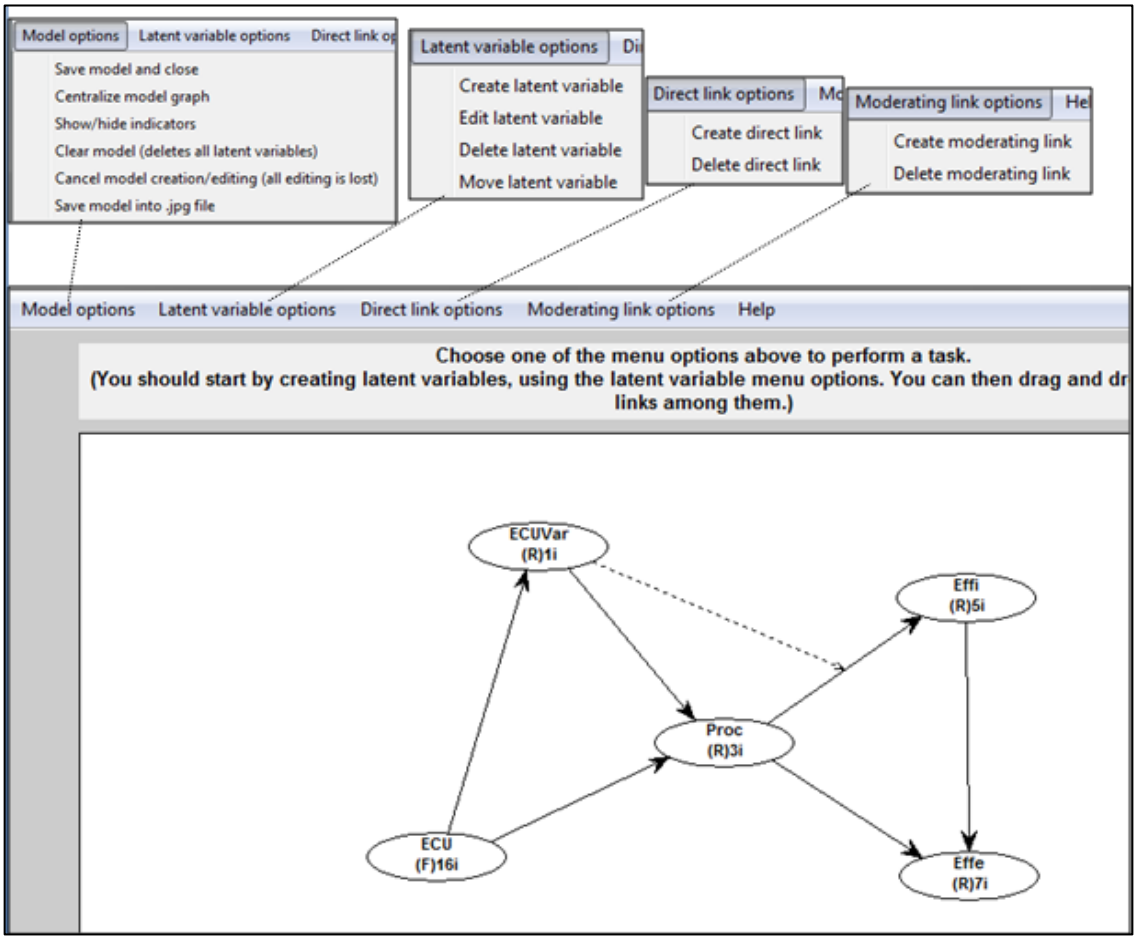
You will define the latent variables by selecting the indicators that are associated with them, and the measurement method used – either formative or reflective. The process of defining the latent variables in a SEM model in this fashion is often called “defining the outer model”, in SEM lingo.

Model links can be of two types, direct and moderating links. Direct links connect pairs of latent variables. Moderating links connect latent variables and direct links; that is, they refer to effects in which a latent variable moderates the relationship between a pair of latent variables. The process of defining model links is often referred to as “defining the inner model”.

F.1. Create or edit the SEM model

The window used to create or edit a model is shown in Figure F.1. A model can be edited if it has been created and saved before as part of a project. While editing or creating a model you can choose from a number of menu options related to overall model functions, latent variable functions, direct link functions, and moderating link functions. As with other windows in this software, there is a help menu option that provides access to this manual in a generic and context-specific manner; both displayed as PDF files. The help menu option also provides links to Web resources.

Figure F.1. Create or edit the SEM model window



A guiding text box is shown at the top of the model editing and creation window. The content of this guiding text box changes depending on the menu option you choose, guiding you through the sub-steps related to each option. For example, if you choose the option “Create latent variable”, the guiding text box will change color, and tell you to select a location for the latent variable on the model graph.

Direct links are displayed as full arrows in the model graph, and **moderating links as dashed arrows**. Each latent variable is displayed in the model graph within an oval symbol, where its name is shown above a combination of alphanumerical characters with this general format: “(F)16i”. The “F” refers to the measurement model; where “F” means formative, and

"R" reflective. The "16i" reflects the number of indicators of the latent variable, which in this case is 16.

Save model and close. This option saves the model within the project, and closes the model editing and creation window. This option does not, however, save the project file. That is, the project file has to be saved for a model to be saved as part of it. This allows you to open a project file, change its model, run a SEM analysis, and discard all that you have done, if you wish to do so, reverting back to the previous project file.

Centralize model graph. This option centralizes the model graph, and is useful when you are building complex models and, in the process of doing so, end up making the model visually unbalanced. For example, you may move variables around so that they are all accidentally concentrated on the left part of the screen. This option corrects that by automatically redrawing all symbols in the model graph so that the center of the model graph coincides with the center of the model screen.

Show/hide indicators. This option shows or hides the list of indicators for each latent variable. The indicators are shown on a vertical list next to each latent variable, and without the little boxes that are usually shown in other SEM software. This display option is used to give the model graph a cleaner look. It also has the advantage that it saves space in the model graph for latent variables. Normally you will want to keep the indicators hidden, except when you are checking whether the right indicators were selected for the right latent variables. That is, normally you will show the indicators to perform a check, and then hide them during most of the model building process.

Clear model (deletes all latent variables). This option deletes all latent variables, essentially clearing the model. Given that choosing this option by mistake can potentially cause some serious loss of work (not to mention some major user aggravation), the software shows a dialog box asking you to confirm that you want to clear the model before it goes ahead and deletes all latent variables. Even if you choose this option by mistake, and confirm your choice also by mistake (a double mistake), you can still undo it by choosing the option "Cancel model creation/editing (all editing is lost)" immediately after clearing the model.

Cancel model creation/editing (all editing is lost). This option cancels the model creation or editing, essentially undoing all of the model changes you have made.

Save model into .jpg file. This option allows you to save the model graph into a .jpg file. You will be asked to select the file name and the folder where the file will be saved. After saved, this file can then be viewed and edited with standard picture viewers, as well as included as a picture into reports in other files (e.g., a Word file). Users can also generate model graph files by copying the model screen into a picture-editing application (e.g., Paint), cropping it to leave out unnecessary or unneeded areas, saving it into a picture file (e.g., .jpg or .png), and then importing that file into reports.

Create latent variable. This option allows you to create a latent variable, and is discussed in more detail below. Once a latent variable is created it can be dragged and dropped anywhere within the window that contains the model.

Edit latent variable. This option allows you to edit a latent variable that has already been created, and thus that is visible on the model graph.

Delete latent variable. This option allows you to delete an existing latent variable. All links associated with the latent variable are also deleted.

Move latent variable. This option is rarely used since, once a latent variable is created, it can be easily dragged and dropped with the pointing device (e.g., mouse) anywhere within the

window that contains the model. This option is a carryover from a previous version, maintained for consistency and for those users who still want to use it. It allows a user to move a latent variable across the model by first clicking on the variable and then on the destination position.

Create direct link. This option allows you to create a direct link between one latent variable and another. The arrow representing the link points from the predictor latent variable to the criterion latent variable. Direct links are usually associated with direct cause-effect hypotheses; testing a direct link's strength (through the calculation of a path coefficient) and statistical significance (through the calculation of a P value) is equivalent to testing a direct cause-effect hypothesis.

Delete direct link. This option allows you to delete an existing direct link. You will click on the direct link that you want to delete, after which the link will be deleted.

Create moderating link. This option allows you to create a link between a latent variable and a direct link. When the underlying algorithm used for outer model estimation is PLS Regression or one of the Factor-Based PLS algorithms, both formative and reflective latent variables can be part of moderating links. Arguably this is not possible with the PLS modes M, A and B (see Lohmöller, 1989), which are usually the ones implemented through other PLS-based SEM software tools. Moderating links are typically associated with moderating cause-effect hypotheses, or interaction effect hypotheses. Testing a moderating link's strength (through the calculation of a path coefficient) and statistical significance (through the calculation of a P value) is equivalent to testing a moderating cause-effect or interaction effect hypothesis. **Moderating links should be used with moderation (no pun intended)**, because they may introduce multicollinearity into the model, and also because they tend to add nonlinearity to the model. By introducing multicollinearity into the model they may make some model parameter estimates unstable and biased.

Delete moderating link. This option allows you to delete an existing moderating link. You will click on the moderating link that you want to delete, after which the link will be deleted.

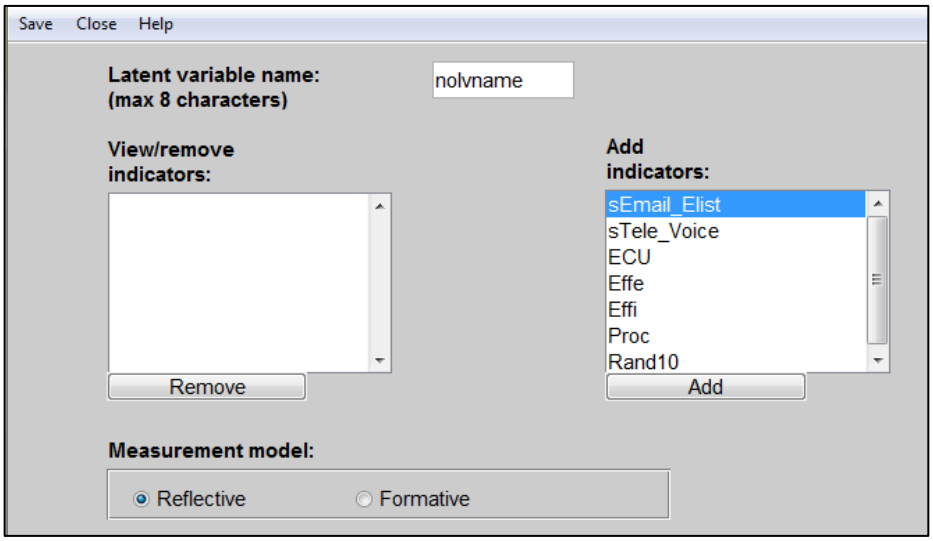
After you create a model and choose the option "Save model and close" **a wait bar will be displayed on the screen telling you that the SEM model structure is being created**. This is an important sub-step where a number of checks are made. In this sub-step, if there are any moderating links in the model, new latent variables are created to store information about those moderating effects using a product-indicator procedure described and validated by Chin et al. (2003). The more moderating links a model has, the longer this sub-step will take. In models where only reflective variables are involved in a moderating link, typically this sub-step will not take longer than a few seconds. Moderating links with formative variables may lead to longer wait times, because formative variables are usually more complex, frequently with significantly more indicators than reflective variables.

Instead of the product-indicator approach described by Chin et al. (2003), one can use an alternative two-stage approach. In the first stage of this alternative approach, the latent variables that are part of a moderating relationship will be added to the model as new indicators. This can be done via the options "**Add one or more latent variable (a.k.a. factor) scores as new standardized indicators**" or "**Add all latent variable (a.k.a. factor) scores as new standardized indicators**", which are available under the "**Modify**" menu options. In the second stage, the new one-indicator latent variables will be used in the definition of a moderating relationship.

F.2. Create or edit latent variable

The latent variable creation window is show in Figure F.2, and it is virtually identical to the latent variable editing window. The latent variable will appear in the model graph as soon as you click on the menu option under "Save", which saves the latent variable and closes the latent variable creation or editing window. A latent variable is not saved as part of a project until the model is saved as part of the project and the project file is saved.

Figure F.2. Create latent variable window



You create a latent variable by entering a name for it, which must have no more than 8 characters, but to which not many other restrictions apply. The latent variable name may contain letters, numbers, and even special characters such as "@" or "$". It cannot contain the special character "*", however, because this character is used later by this software in selected outputs to indicate that a latent variable is associated with a moderating effect. After entering a name for a latent variable, you then select the indicators that make up the latent variable, and define the measurement model as reflective or formative.

A **reflective latent variable** is one in which all the indicators are expected to be highly correlated with one another, and with the latent variable itself. For example, the answers to certain question-statements by a group of people, measured on a 1 to 7 scale (1=strongly disagree; 7=strongly agree) and answered after a meal, are expected to be highly correlated with the latent variable "satisfaction with a meal". Among question-statements that would arguably fit this definition are the following two: "I am satisfied with this meal", and "After this meal, I feel full". Therefore, the latent variable "satisfaction with a meal", can be said to be reflectively measured through two indicators. Those indicators store answers to the two question-statements. This latent variable could be represented in a model graph as "Satisf", and the indicators as "Satisf1" and "Satisf2". Notwithstanding this simplified example, users should strive to have more than two indicators be latent variable; the more indicators, the better, since the number of indicators is inversely related to the amount of measurement error (Kock, 2014; Nunnally, 1978; Nunnally & Bernstein, 1994).

A **formative latent variable** is one in which the indicators are expected to measure certain attributes of the latent variable, but the indicators are not expected to be highly correlated with

the latent variable itself, because they (i.e., the indicators) are not expected to be highly correlated with one another. For example, let us assume that the latent variable "Satisf" ("satisfaction with a meal") is now measured using the two following question-statements: "I am satisfied with the main course" and "I am satisfied with the dessert". Here, the meal comprises the main course, say, filet mignon; and a dessert, such as a fruit salad. Both main course and dessert make up the meal (i.e., they are part of the same meal) but their satisfaction indicators are not expected to be highly correlated with each other. The reason is that some people may like the main course very much, and not like the dessert. Conversely, other people may be vegetarians and hate the main course, but may like the dessert very much.

If the indicators are not expected to be highly correlated with one another, they cannot be expected to be highly correlated with their latent variable's score. Here is a general rule of thumb that can be used to decide if a latent variable is reflectively or formatively measured. If the indicators are expected to be highly correlated, and are redundant in their meaning, then the measurement model should be set as reflective. If the indicators are not expected to be highly correlated, and are clearly not redundant in meaning (they measure different facets of the same construct), even though they clearly refer to the same latent variable, then the measurement model should be set as formative.

Setting a latent variable as formative or reflective affects the calculation of model parameters only with the PLS Mode B algorithm, or with algorithms that employ the PLS Mode B algorithm or variations of it (e.g., PLS Mode B Basic, PLS Mode M). With other algorithms, setting a latent variable as formative or reflective is still recommended, as it helps the user interpret outputs and conduct certain assessments (e.g., validity assessments, discussed later in this manual).

Formative measurement has been facing increasing criticism, particularly since the late 1990s; see Edwards (2011) for a particularly critical and cogent discussion. Given this growing criticism, it is recommended that the Cronbach's alpha coefficients associated with formative latent variables be equal to or greater than 0.6, for reasons related to measurement error theory (Kock, 2014; Nunnally, 1978; Nunnally & Bernstein, 1994). Since loadings tend to be relatively low with formative latent variables (and weights relatively high), reliability measures (such as the Cronbach's alpha coefficient) tend to also be relatively low. Nevertheless, Cronbach's alpha coefficients equal to or greater than 0.6 can be achieved by increasing the number of indicators used in formative measurement.

G. Step 5: Perform the SEM analysis and view the results

Step 5 performs the SEM analysis based on the model created in Step 4. After you click on the button to perform the SEM analysis, the software will show a wait bar. This wait bar will update you on the progress of the SEM analysis, which usually will take only a few seconds or less for simple to moderately complex models. As soon as the SEM analysis is completed, the software will show the results in graphical format on a window. That window also has menu options that allow you to view more details about the results, including some that are not shown on the graph (e.g., reliability measures), and also save the results into tab-delimited text files.

Collinearity is estimated before the SEM analysis is run. If collinearity appears to be too high, users are warned about it. A table with estimated latent variable correlations is shown, allowing users to identify the possible offending latent variables. If users so choose, they can proceed with the analysis anyway, but in most (not all) cases the full collinearity (a.k.a. multicollinearity) measures will confirm that collinearity is too high in their models for the analysis results to be considered credible.

Measurement error and composite weights are estimated before the SEM analysis is run, whenever Factor-Based PLS algorithms are used. Measurement error and composite weights play a key role in these algorithms. If at least one measurement error weight is greater than the corresponding composite weight, the user is warned about possible unreliability of results. This happens usually when at least one of the Cronbach's alpha coefficients associated with the latent variables is lower than 0.5. Foundational aspects of the Factor-Based PLS algorithms are discussed by Kock (2014), who lays out the mathematical basis of these algorithms, from which the importance of measurement error and composite weights can be gleaned.

New options become available from the main window after Step 5 is completed, under the "Modify" menu option. These options allow users to add one or more latent variable scores to the model as new standardized indicators, and also to add all latent variable scores as new indicators. Adding one or more latent variable scores at a time may be advisable in certain cases, such as in hierarchical analyses using selected latent variable scores as indicators at each level. In these cases, adding all latent variable scores at once may soon clutter the set of indicators available to be used in the SEM model.

The option of adding latent variable scores to the model as new standardized indicators is useful in the removal of outliers, through the use of restricted ranges for latent variable scores, particularly for outliers that are clearly visible on the plots depicting associations among latent variables. As briefly mentioned earlier, this option is also useful in hierarchical analyses, where users define second-order (and higher order) latent variables, and then conduct analyses with different models including latent variables of different orders.

This software uses algorithms that are fairly computing intensive, in some cases employing multiple checks and optimization sub-algorithms in each sub-step. Therefore the speed with which the analysis is conducted may be a little slower than that of some other publicly available SEM software. The differences in speed are not significant though, and normally the results generated by this software are more complete, and in many cases more reliable. For example, this software calculates model fit and quality indices, as well as P values for most of its parameter estimates. Publicly available PLS-based SEM software usually do not provide those measures.

Some model elements may reduce the speed of the SEM analysis more than others. These are: formative latent variables with many indicators and, more generally, latent variables with

many indicators (even if they are reflective); moderating effects, particularly if they associate latent variables with many indicators; setting the number of resamples for Bootstrapping or Blindfolding as 200 or higher; and using Jackknifing as the resampling method, if the sample size is larger than 200.

In Jackknifing, the number of resamples equals the sample size, which is why using Jackknifing as the resample method may reduce the speed of the SEM analysis with relatively large samples. Generating resamples and running calculations on them is one of the most computing intensive sub-steps of the SEM analysis. However, Jackknifing often produces more stable parameter estimates with warped analysis. So there is a tradeoff between speed and reliability when warping algorithms are being used. This tradeoff may tip the balance in favor of using Jackknifing, alone or in addition to Bootstrapping or Blindfolding, even if the user has to wait longer for the results.

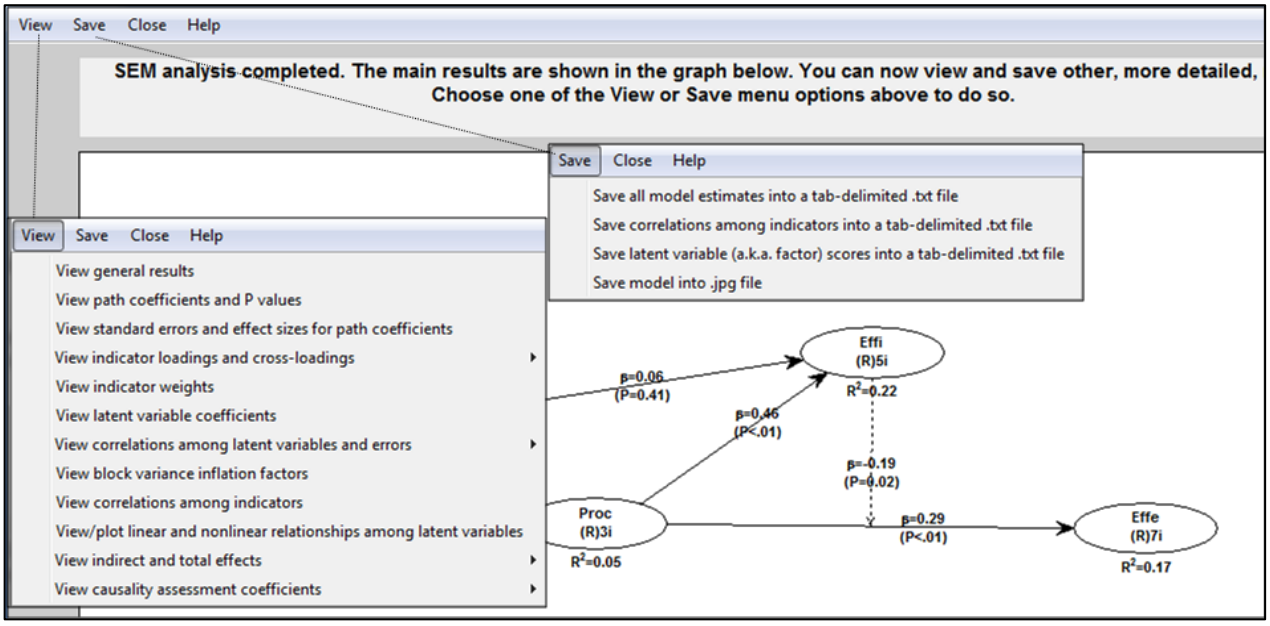
An alternative is the use of the “stable” quasi-parametric methods: Stable1, Stable2, and Stable3. This alternative is highly recommended, particularly with the Stable3 method, the software’s default. As their name implies, these methods yield stable coefficients. They also provide fairly accurate estimates of standard errors, which are used in the calculation of P values. These methods do not actually generates resamples, so calling them resampling methods is done here for simplicity in the grouping of settings options. Because no resamples are generated, these are rather efficient methods from a computing load perspective. These methods can be particularly useful in the analysis of large datasets, as in these cases creating resamples can be computationally very taxing. With the emergence of the concept of “big data”, the need to analyze large datasets is becoming increasingly common.

H. View and save results

As soon as the SEM analysis is completed, the software shows the results in graphical format on a window, which also contains a number of menu options that allow you to view and save more detailed results (see Figure H.1.1). The graph with the results shows path coefficients, respective P values, and R-squared coefficients.

The "Save" menu options allow users to save all of the results that they can view, with the majority of those results saved under the option to save all model estimates into a tab-delimited text file. Additionally, users can save the factor scores calculated for each latent variable. These can be useful in some specialized applications; e.g., users may want to generate customized graphs based on those scores.

Figure H.1.1. View and save results window


View Save Close Help
SEM analysis completed. The main results are shown in the graph below. You can now view and save other, more detailed,
Choose one of the View or Save menu options above to do so.
Save Close Help
Save all model estimates into a tab-delimited .txt file
Save correlations among indicators into a tab-delimited .txt file
Save latent variable (a.k.a. factor) scores into a tab-delimited .txt file
Save model into .jpg file
View Save Close Help
View general results
View path coefficients and P values
View standard errors and effect sizes for path coefficients
View indicator loadings and cross-loadings
View indicator weights
View latent variable coefficients
View correlations among latent variables and errors
View block variance inflation factors
View correlations among indicators
View/plot linear and nonlinear relationships among latent variables
View indirect and total effects
View causality assessment coefficients
Effi (R)5i
R²=0.22
β=0.06 (P=0.41)
β=0.46 (P<.01)
β=-0.19 (P=0.02)
Proc (R)3i
R²=0.05
β=0.29 (P<.01)
Effe (R)7i
R²=0.17

Just to be clear, the "factor" scores are the latent variable scores; even though classic PLS algorithms approximate latent variables though composites, not factors. This is generally perceived as a limitation of classic PLS algorithms (Kock, 2014; 2014d), which is addressed through the Factor-Based PLS algorithms. The latter, Factor-Based PLS algorithms, estimate latent variables through the estimation of the true factors. The term "factor" is often used when we refer to latent variables, in the broader context of SEM analyses in general. The reason is that factor analysis, from which the term "factor" originates, can be seen as a special case of SEM analysis.

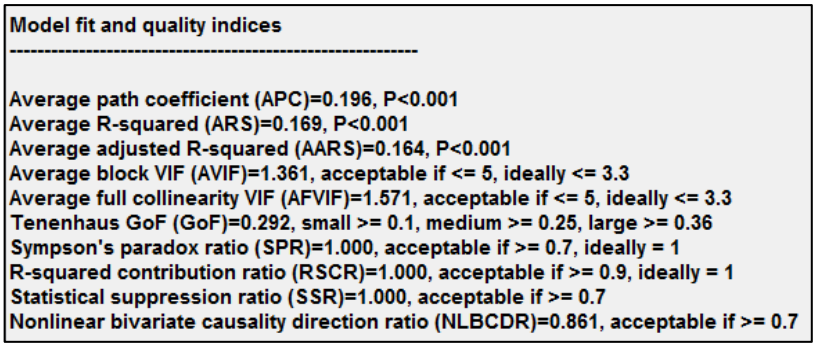
The path coefficients are noted as beta coefficients. "Beta coefficient" is another term often used to refer to path coefficients in PLS-based SEM analyses; this term is commonly used in multiple regression analyses. The P values are displayed below the path coefficients, within parentheses. The R-squared coefficients are shown below each endogenous latent variable (i.e., a latent variable that is hypothesized to be affected by one or more other latent variables), and reflect the percentage of the variance in the latent variable that is explained by the latent variables that are hypothesized to affect it. To facilitate the visualization of the results, the path

coefficients and P values for moderating effects are shown in a way similar to the corresponding values for direct effects, namely next to the arrows representing the effects.

H.1. View general results

General SEM analysis results include: the version of WarpPLS used in the SEM analysis; project file details, such as the project file name and when the file was last saved; model fit and quality indices (shown in Figure H.1.2), which are discussed in more detail below; and general model elements, such as the algorithm and resampling method used in the SEM analysis.

Figure H.1.2. General results window


Model fit and quality indices
Average path coefficient (APC)=0.196, P<0.001
Average R-squared (ARS)=0.169, P<0.001
Average adjusted R-squared (AARS)=0.164, P<0.001
Average block VIF (AVIF)=1.361, acceptable if <= 5, ideally <= 3.3
Average full collinearity VIF (AFVIF)=1.571, acceptable if <= 5, ideally <= 3.3
Tenenhaus GoF (GoF)=0.292, small >= 0.1, medium >= 0.25, large >= 0.36
Sympson's paradox ratio (SPR)=1.000, acceptable if >= 0.7, ideally = 1
R-squared contribution ratio (RSCR)=1.000, acceptable if >= 0.9, ideally = 1
Statistical suppression ratio (SSR)=1.000, acceptable if >= 0.7
Nonlinear bivariate causality direction ratio (NLBCDR)=0.861, acceptable if >= 0.7

Under the project file details, both the raw data path and file are provided. Those are provided for completeness, because once the raw data is imported into a project file, it is no longer needed for the analysis. **Once a raw data file is read, it can even be deleted without any effect on the project file, or the SEM analysis**.

Ten global **model fit and quality indices** are provided: **average path coefficient (APC)**, **average R-squared (ARS)**, **average adjusted R-squared (AARS)**, **average block variance inflation factor (AVIF), average full collinearity VIF (AFVIF), Tenenhaus GoF (GoF)**, **Simpson's paradox ratio (SPR)**, **R-squared contribution ratio (RSCR)**, **statistical suppression ratio (SSR)**, and **nonlinear bivariate causality direction ratio (NLBCDR)**.

For the APC, ARS, and AARS, P values are also provided. These P values are calculated through a process that involves resampling estimations coupled with corrections to counter the standard error compression effect associated with adding random variables, in a way analogous to Bonferroni corrections (Rosenthal & Rosnow, 1991). This is necessary since the model fit and quality indices are calculated as averages of other parameters.

The interpretation of the model fit and quality indices depends on the goal of the SEM analysis. If the goal is to only test hypotheses, where each arrow represents a hypothesis, then the model fit and quality indices are, as a whole, of less importance. However, if the goal is to find out whether one model has a better fit with the original data than another, then the model fit and quality indices are a useful set of measures related to model quality. **When assessing the model fit with the data**, several criteria are recommended. These criteria are discussed below, together with the discussion of the model fit and quality indices.

APC, ARS and AARS. Typically the addition of new latent variables into a model will increase the ARS, even if those latent variables are weakly associated with the existing latent variables in the model. However, that will generally lead to a decrease in the APC, since the path coefficients associated with the new latent variables will be low. Thus, the APC and ARS will counterbalance each other, and will only increase together if the latent variables that are added to the model enhance the overall predictive and explanatory quality of the model. The AARS is generally lower than the ARS for a given model. The reason is that it averages adjusted R-

squared coefficients (Theil, 1958; Wooldridge, 1991), which themselves correct for spurious increases in R-squared coefficients due to predictors that add no explanatory value in each latent variable block. It is recommended that the **P values for the APC, ARS and AARS all be equal to or lower than 0.05**; that is, significant at the 0.05 level. A more relaxed rule would be that the **P values for the APC and ARS only be equal to or lower than 0.05**.

AVIF and AFVIF. The AVIF index will increase if new latent variables are added to the model in such a way as to add vertical collinearity in the model's latent variable blocks. The AFVIF index will increase if new latent variables are added to the model in such a way as to add full collinearity into the model (i.e., either vertical or lateral collinearity; see Kock & Lynn, 2012). Full collinearity is often referred to as "muticollinearity". High AVIF and AFVIF values may result from the inclusion of new latent variables that overlap in meaning with existing latent variables. It is generally undesirable to have different latent variables in the same model that measure the same underlying construct; those should be combined into one single latent variable. Thus, the AVIF and AFVIF indices bring in new dimensions that add to a comprehensive assessment of a model's overall predictive and explanatory quality. Because of the way in which these indices are calculated (for more details, see: Kock & Lynn, 2012), the AFVIF is not sensitive to variations in collinearity due to the use of nonlinear algorithms. The AVIF, on the other hand, is sensitive to the use of nonlinear algorithms. Therefore it is **recommended that both indices, AVIF and AFVIF, be reported** in studies, as they are not redundant indices. It is recommended (ideally) that **both the AVIF and AFVIF be equal to or lower than 3.3, particularly in models where most of the variables are measured through two or more indicators**. A more relaxed (acceptable) criterion **is that both indices be equal to or lower than 5, particularly in models where most variables are single-indicator variables (and thus not "true" latent variables)**. The reason for these differences in criteria in different contexts is that PLS-based SEM algorithms in general tend to be particularly effective at reducing collinearity (Kock & Lynn, 2012), but only if multiple indicators are available to be aggregated in the calculation of latent variable scores.

GoF. Similarly to the ARS, the GoF index, referred to as "Tenenhaus GoF" in honor of Michel Tenenhaus, is a measure of a model's explanatory power. Tenenhaus et al. (2005) defined the GoF as the square root of the product between what they refer to as the average communality index and the ARS. The communality index for a given latent variable is defined as the sum of the squared loadings for that latent variable, each loading associated with an indicator, divided by the number of indicators. The average communality index for a model is defined similarly, and takes all latent variables into account in its calculation. The loadings referred to here are the unrotated loadings, which are available from the structure loadings and cross-loadings table. It is also worth noting that the definition of the communality index used by Tenenhaus et al. (2005) does not match the typical definition of communality, at least as it is normally stated in the context of factor analysis. As noted by Wetzels et al. (2009), the average variance extracted (AVE) for each latent variable equals the corresponding communality index. Wetzels et al. (2009) also proposed the **following thresholds for the GoF: small if equal to or greater than 0.1, medium if equal to or greater than 0.25, and large if equal to or greater than 0.36**. They did so by assuming a minimum acceptable average AVE of 0.5, and using Cohen's (1988) thresholds for small, medium, and large effect sizes. A value lower than 0.1 for the GoF suggests that the explanatory power of a model may be too low to be considered acceptable.

SPR. The SPR index is a measure of the extent to which a model is free from Simpson's paradox instances (Pearl, 2009; Wagner, 1982). An instance of Simpson's paradox occurs when a path coefficient and a correlation associated with a pair of linked variables have different signs. A Simpson's paradox instance is a possible indication of a causality problem, suggesting that a hypothesized path is either implausible or reversed. The SPR index is calculated by dividing the number of paths in a model that are not associated with Simpson's paradox instances by the total number of paths in the model. At the time of this writing the SPR was an experimental index, and thus the following recommendations should also be treated as experimental. **Ideally the SPR should equal 1**, meaning that there are no instances of Simpson's paradox in a model; **acceptable values of SPR are equal to or greater than 0.7**, meaning that at least 70 percent of the paths in a model are free from Simpson's paradox.

RSCR. The RSCR index is a measure of the extent to which a model is free from negative R-squared contributions, which occur together with Simpson's paradox instances (Pearl, 2009; Wagner, 1982). When a predictor latent variable makes a negative contribution to the R-squared of a criterion latent variable (note: the predictor points at the criterion), this means that the predictor is actually *reducing* the percentage of variance explained in the criterion. Such a reduction takes into consideration the contributions of all predictors plus that of the residual. This index is similar to the SPR. The key difference is that it is calculated based on the actual values of the R-squared contributions, not on the number of paths where these contributions have specific signs. The RSCR index is calculated by dividing the sum of positive R-squared contributions in a model by the sum of the absolute R-squared contributions (be they negative or positive) in the model. At the time of this writing the RSCR was an experimental index, and thus the following recommendations should also be treated as experimental. **Ideally the RSCR should equal 1**, meaning that there are no negative R-squared contributions in a model; **acceptable values of RSCR are equal to or greater than 0.9**, meaning that the sum of positive R-squared contributions in a model makes up at least 90 percent of the total sum of the absolute R-squared contributions in the model.

SSR. The SSR index is a measure of the extent to which a model is free from statistical suppression instances (MacKinnon et al., 2000). An instance of statistical suppression occurs when a path coefficient is greater, in absolute terms, than the corresponding correlation associated with a pair of linked variables. Like a Simpson's paradox instance, a statistical suppression instance is a possible indication of a causality problem (Spirtes et al., 1993), suggesting that a hypothesized path may be either implausible or reversed. The SSR index is calculated by dividing the number of paths in a model that are not associated with medium or greater statistical suppression instances by the total number of paths in the model. A medium or greater statistical suppression instance is characterized by an absolute path-correlation ratio that is greater than 1.3. At the time of this writing the SSR was an experimental index, and thus the following recommendation should also be treated as experimental. **Acceptable values of SSR are equal to or greater than 0.7**, meaning that at least 70 percent of the paths in a model are free from statistical suppression.

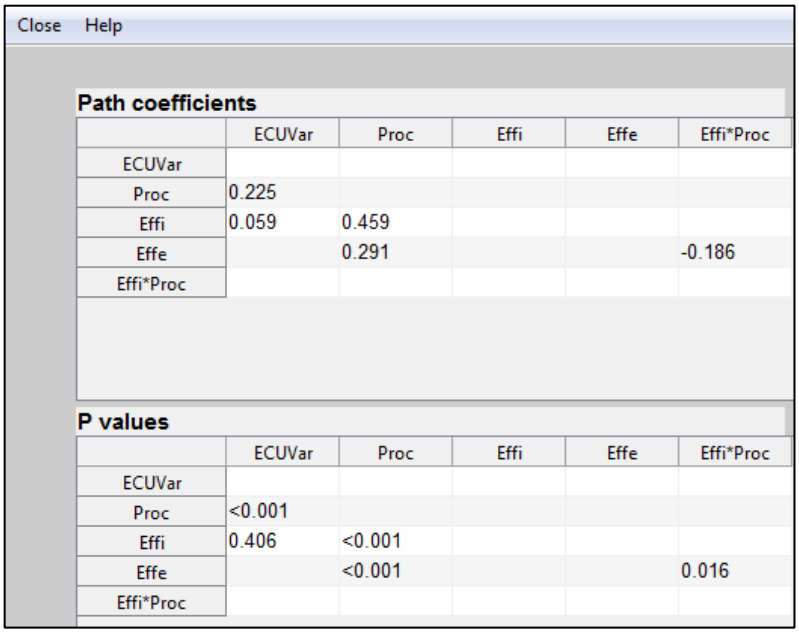
NLBCDR. One interesting property of nonlinear algorithms is that bivariate nonlinear coefficients of association vary depending on the hypothesized direction of causality. That is, they tend to be stronger in one direction than the other, which means that the residual (or error) is greater when the hypothesized direction of causality is in one way or another. As such, they can be used, together with other coefficients, as partial evidence in support or against hypothesized causal links. The NLBCDR index is a measure of the extent to which bivariate nonlinear

coefficients of association provide support for the hypothesized directions of the causal links in a model. The NLBCDR index is calculated by dividing the number of path-related instances in a model where the support for the reversed hypothesized direction of causality is more than weak by the total number of path-related instances involved in this test (this is discussed in more detail later). All of the available nonlinear algorithms are used in this test. Therefore the total number of path-related instances involved in this test is greater than the total number of paths. At the time of this writing the NLBCDR was an experimental index, and thus the following recommendation should also be treated as experimental. **Acceptable values of NLBCDR are equal to or greater than 0.7**, meaning that in at least 70 percent of path-related instances in a model the support for the reversed hypothesized direction of causality is weak or less. Here "less" may mean that the support for reversed hypothesized direction of causality is less than weak (e.g., neutral), or that the hypothesized direction of causality is supported.

H.2. View path coefficients and P values

Path coefficients and respective P values are shown together, as can be seen in Figure H.2. Each path coefficient is displayed in one cell, where the column refers to the predictor latent variable and the row to the criterion. For example, let us consider the case in which the cell shows 0.225, the column refers to the latent variable “ECUVar”, and the row to the latent variable “Proc”. This means that the path coefficient associated with the arrow that points from “ECUVar” to “Proc” is 0.225.

Figure H.2. Path coefficients and P values window



Since the results refer to standardized variables, a path coefficient of 0.225 means that, in a linear analysis, a 1 standard deviation variation in “ECUVar” leads to a 0.225 standard deviation variation in “Proc”. In a nonlinear analysis, the meaning is generally the same, except that it applies to the overall linear trend of the transformed (or warped) relationship. However, it is important to note that, in nonlinear relationships the path coefficient at each point of a curve varies. In nonlinear relationships, the path coefficient is given by the first derivative of the nonlinear function that describes the relationship.

The P values shown are calculated through one of several methods available, and are thus method-specific; i.e., they change based on the P value calculation method chosen. In the calculation of P values, a one-tailed test is generally recommended if the coefficient is assumed to have a sign (positive or negative), which should be reflected in the hypothesis that refers to the corresponding association (Kock, 2014d). Hence this software reports one-tailed P values for path coefficients; from which two-tailed P values can be easily obtained if needed (Kock, 2014d).

One puzzling aspect of many publicly available PLS-based SEM software systems is that they have historically avoided providing P values, instead providing standard errors and T values, and leaving the users to figure out what the corresponding P values are. Often users have to resort to

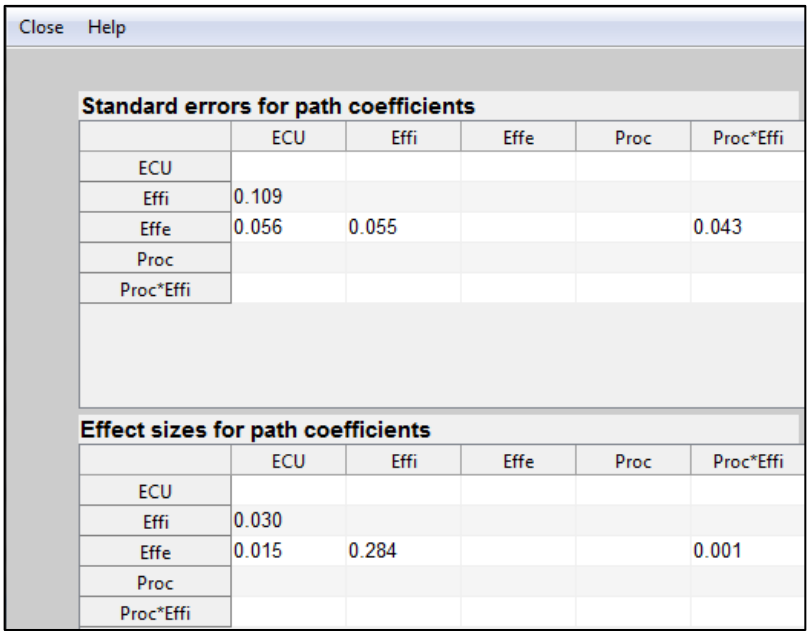
tables relating T to P values, or other software (e.g., Excel), to calculate P values based on T values.

This is puzzling because typically research reports will provide P values associated with path coefficients, which are more meaningful than T values for hypothesis testing purposes. This is due to the fact that P values reflect not only the strength of the relationship (which is already provided by the path coefficient itself) but also the power of the test, which increases with sample size. The larger the sample size, the lower a path coefficient has to be to yield a statistically significant P value.

H.3. View standard errors and effect sizes for path coefficients

Standard errors and effect sizes for path coefficients are provided in two tables where one standard error and effect size is provided for each path coefficient (see Figure H.3). The effect sizes provided are similar to Cohen's (1988) *f*-squared coefficients. Standard errors and effect sizes are provided in the same order as the path coefficients, so that users can easily visualize them; and, in certain cases, use them to perform additional analyses.

Figure H.3. Standard errors and effect sizes for path coefficients window



Even though the effect sizes provided are similar to Cohen's (1988) *f*-squared coefficients, they are calculated using a different procedure. The reason for this is that the stepwise regression procedure proposed by Cohen (1988) for the calculation of *f*-squared coefficients is generally not compatible with PLS-based SEM algorithms. The removal of predictor latent variables in latent variable blocks, used in the stepwise regression procedure proposed by Cohen (1988), tends to cause changes in the weights linking latent variable scores and indicators, thus biasing the effect size measures.

The effect sizes are calculated by this software as the absolute values of the individual contributions of the corresponding predictor latent variables to the R-squared coefficients of the criterion latent variable in each latent variable block. **With the effect sizes users can ascertain whether the effects indicated by path coefficients are small, medium, or large. The values usually recommended are 0.02, 0.15, and 0.35**; respectively (Cohen, 1988). **Values below 0.02 suggest effects that are too weak to be considered relevant from a practical point of view, even when the corresponding P values are statistically significant**; a situation that may occur with large sample sizes.

Additional types of analyses that may be conducted with standard errors are tests of the significance of any mediating effects using the approach discussed by Kock (2013). This approach consolidates the approaches discussed by Preacher & Hayes (2004), for linear relationships; and Hayes & Preacher (2010), for nonlinear relationships. The latter, discussed by

Hayes & Preacher (2010), assumes that nonlinear relationships are force-modeled as linear; which means that the equivalent test using this software would use warped coefficients with the earlier linear approach discussed by Preacher & Hayes (2004). Again, for the consolidated version of these approaches, see Kock (2013). The classic approach used for testing mediating effects is discussed by Kock (2011b). This approach is a concise version of Baron & Kenny's (1986) classic approach, which does not rely on standard errors.

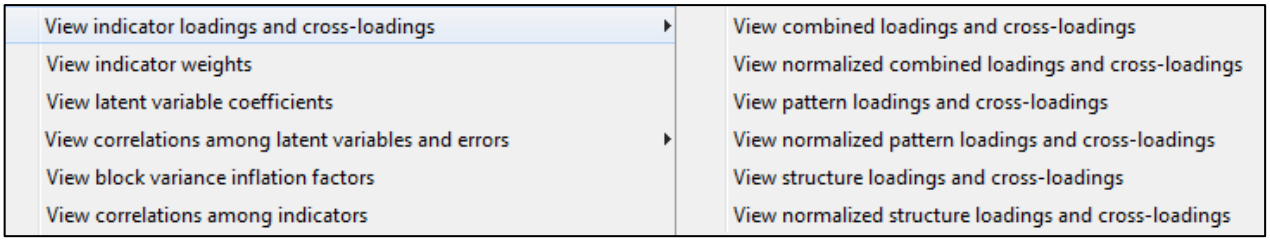
An alternative approach to the analysis of mediating effects, which is arguably much less time-consuming and prone to error than the approaches mentioned above, would be to rely on the estimation of indirect effects. These indirect effects and related P values are automatically calculated by the software, and allow for the test of multiple mediating effects at once, including effects with more than one mediating variable. Kock & Gaskins (2014) provide an empirical illustration of the use of this approach. Indirect and total effects are discussed in more detail later.

Another type of analysis that can employ standard errors for path coefficients is what is often referred to as a multi-group analysis, where path and measurement model coefficients (usually weights) can be compared. One of the main goals of this type of analysis is to compare pairs of path coefficients for identical models but based on different samples. An example would be the analysis of the same model but with data collected in two different countries. See Kock (2013) for a more detailed discussion on the use of effect sizes and other coefficients generated by this software on advanced mediating effects tests, comprehensive multi-group analyses, and measurement model assessments.

H.4. View indicator loadings and cross-loadings

The "View indicator loadings and cross-loadings" menu options (see Figure H.4.1) allow users to view combined loadings and cross-loadings, normalized combined loadings and cross-loadings, pattern loadings and cross-loadings, normalized pattern loadings and cross-loadings, structure loadings and cross-loadings, and normalized structure loadings and cross-loadings.

Figure H.4.1. Indicator loadings and cross-loadings options


View indicator loadings and cross-loadings
View indicator weights
View latent variable coefficients
View correlations among latent variables and errors
View block variance inflation factors
View correlations among indicators
View combined loadings and cross-loadings
View normalized combined loadings and cross-loadings
View pattern loadings and cross-loadings
View normalized pattern loadings and cross-loadings
View structure loadings and cross-loadings
View normalized structure loadings and cross-loadings

Combined loadings and cross-loadings are shown in a window, as illustrated in Figure H.4.2. The same is true for other combinations of loadings and cross-loadings, which are shown in similar windows. Combined loadings and cross-loadings are provided in a table with each cell referring to an indicator-latent variable link. Latent variable names are listed at the top of each column, and indicator names at the beginning of each row. In this table, the loadings are from a structure matrix (i.e., unrotated), and the cross-loadings from a pattern matrix (i.e., rotated). Indicator types, as defined, are also provided – reflective or formative.

Figure H.4.2. Combined loadings and cross-loadings window

	ECUVar	Proc	Effi	Effe	Type (as defined)	SE	P value
ECUVari	(1.000)	0.000	0.000	0.000	Reflective	0.050	<0.001
Proc1	0.000	(0.844)	0.039	-0.030	Reflective	0.051	<0.001
Proc2	-0.040	(0.885)	-0.081	0.130	Reflective	0.051	<0.001
Proc3	0.044	(0.814)	0.047	-0.111	Reflective	0.052	<0.001
Effi1	-0.006	-0.014	(0.896)	-0.019	Reflective	0.051	<0.001
Effi2	-0.065	0.035	(0.884)	-0.126	Reflective	0.051	<0.001
Effi3	0.005	0.052	(0.821)	-0.026	Reflective	0.052	<0.001
Effi4	0.079	-0.117	(0.813)	-0.011	Reflective	0.052	<0.001
Effi5	-0.008	0.043	(0.796)	0.200	Reflective	0.052	<0.001
Effe1	-0.041	0.060	-0.050	(0.931)	Reflective	0.051	<0.001
Effe2	0.032	-0.007	-0.051	(0.947)	Reflective	0.050	<0.001
Effe3	0.006	0.024	0.080	(0.831)	Reflective	0.051	<0.001
Effe4	0.007	0.018	-0.038	(0.952)	Reflective	0.050	<0.001
Effe5	0.036	-0.075	0.006	(0.917)	Reflective	0.051	<0.001
Effe6	0.032	-0.026	-0.066	(0.906)	Reflective	0.051	<0.001
Effe7	-0.074	0.007	0.132	(0.894)	Reflective	0.051	<0.001

In the combined loadings and cross-loadings window, **since loadings are from a structure matrix, and unrotated, they are always within the -1 to 1 range**. With some exceptions, which are discussed below, this obviates the need for a normalization procedure to avoid the presence of loadings whose absolute values are greater than 1. The expectation here is that for reflective latent variables loadings, which are shown within parentheses, will be high; and cross-

loadings will be low. **The type of the latent variable as defined** by the user, namely **reflective or formative**, **is also provided** in this window to facilitate the application of validity and reliability tests. The criteria used in these tests are typically different for formative and reflective latent variables.

P values are provided for indicators associated with all latent variables. These P values are often referred to as validation parameters of a confirmatory factor analysis (Kline, 1998; Schumacker & Lomax, 2004), since they result from a test of a model where the relationships between indicators and latent variables are defined beforehand. Conversely, in an exploratory factor analysis (Ehremberg & Goodhart, 1976), relationships between indicators and latent variables are not defined beforehand, but inferred based on the results of a factor extraction algorithm. The principal components analysis algorithm is one of the most popular of these algorithms, even though it is often classified as outside the scope of classic factor analysis. Confirmatory factor analyses, instead of exploratory factor analyses, are usually conducted in conjunction with SEM analyses.

For research reports, users will typically use the table of combined loadings and cross-loadings provided by this software when describing the convergent validity of their measurement instrument. A measurement instrument has good convergent validity if the question-statements (or other measures) associated with each latent variable are understood by the respondents in the same way as they were intended by the designers of the question-statements. In this respect, two criteria are recommended as the basis for concluding that a measurement model has acceptable convergent validity: that the **P values associated with the loadings be equal to or lower than 0.05**; and that the **loadings be equal to or greater than 0.5** (Hair et al., 1987; 2009).

Indicators for which these criteria are not satisfied may be removed. **This does not apply to formative latent variable indicators, which are assessed in part based on P values associated with indicator weights**. If the offending indicators are part of a moderating effect, then you should consider removing the moderating effect if it does not meet the requirements for formative measurement. Moderating effect latent variable names are displayed on the table as product latent variables (e.g., Effi*Proc).

Moderating effect indicator names are displayed on the table as product indicators (e.g., “Effi1*Proc1”). Long names are reduced to avoid a “crowded” look. High P values for moderating effects, to the point of being non-significant at the 0.05 level, may suggest multicollinearity problems; which can be further checked based on the latent variable coefficients generated by the software, more specifically, the full collinearity VIFs. Some degree of collinearity is to be expected with moderating effects, since the corresponding product variables are likely to be correlated with at least their component latent variables. Moreover, moderating effects add nonlinearity to models, which can in some cases compound multicollinearity problems. Because of these and other related issues, moderating effects should be included in models with caution.

Standard errors are also provided for the loadings, in the column indicated as “SE”, for indicators associated with all latent variables. They can be used in specialized tests. Among other purposes, these standard errors can be used in multi-group analyses, with the same model but different subsamples. In these cases, users may want to compare the measurement models to ascertain equivalence based on loadings and weights, using a multi-group comparison technique such as the one documented by Kock (2013) and Keil et al. (2000), and thus ensure that any observed between-group differences in structural model coefficients are not due to measurement model differences. Keil et al.’s (2000) discussion on multi-group analyses includes an equation

that contains an error; the correct form of the equation is used in Kock's (2013) discussion. The equation in question is for the calculation of a pooled standard error, and is one of the two equations discussed by Kock (2013) in the context of multi-group analyses; the other implements the alternative Satterthwaite method. According to Keil et al. (2000), the original proponent of the pooled standard error equation is Wynne Chin, one of the world's foremost authorities on PLS-based SEM.

Normalized loadings and cross-loadings. Normalized versions of the combined, pattern, and structure loadings and cross-loadings tables are also provided. In windows showing normalized loadings and cross-loadings, a Kaiser normalization is employed to calculate them (Ferguson, 1981; Kaiser, 1958; Ogasawara, 1999). Through a Kaiser normalization, each row of a table of loadings and cross-loadings is divided by the square root of its communality. This has the effect of making the sum of squared values in each row add up to 1.

Using a Kaiser normalization is reasonably standard practice (Ferguson, 1981; Ogasawara, 1999). Sometimes the normalization is followed by a de-normalization, which is not the case with this software. **The normalized values are useful in situations where the PLS Regression algorithm is used and some of the latent variables have only 2 indicators, particularly with respect to options displaying unrotated loadings**. In such cases the unrotated loadings in the combined loadings and cross-loadings window that are associated with each of the 2 indicators are the same, because with the PLS Regression algorithm the inner model does not influence the outer model. A Kaiser normalization will usually make indicator loadings diverge in value in these cases, in a way that is consistent with standard practice and that, some researchers argue, frequently leads to more conservative estimates of loadings and cross-loadings. For a more detailed discussion, and a review of different perspectives on this topic, see Ogasawara (1999).

Pattern loadings and cross-loadings are provided in a table with each cell referring to an indicator-latent variable link. Latent variable names are listed at the top of each column, and indicator names at the beginning of each row. In this table, both the loadings and cross-loadings are from a pattern matrix (i.e., rotated).

Since these loadings and cross-loadings are from a pattern matrix, they are obtained after the transformation of a structure matrix through a widely used oblique rotation frequently referred to as Promax. The structure matrix contains the Pearson correlations between indicators and latent variables, which are not particularly meaningful prior to rotation in the context of measurement instrument validation. **Because an oblique rotation is employed, in some cases loadings may be higher than 1** (Rencher, 1998). This could be a hint that two or more latent variables are collinear, although this may not necessarily be the case; better measures of collinearity among latent variables are the full collinearity VIFs reported with other latent variable coefficients. **In the normalized version of this table, typically there will be no loadings higher than 1**.

The main difference between oblique and orthogonal rotation methods is that the former assume that there are correlations, some of which may be strong, among latent variables. Arguably **oblique rotation methods are the most appropriate in a SEM analysis**, because by definition latent variables are expected to be correlated. Otherwise, no path coefficient would be significant. Technically speaking, it is possible that a research study will hypothesize only neutral relationships between latent variables, which could call for an orthogonal rotation. However, this is rarely, if ever, the case.

Structure loadings and cross-loadings are provided in a table with each cell referring to an indicator-latent variable link. Latent variable names are listed at the top of each column, and indicator names at the beginning of each row. In this table, both the loadings and cross-loadings

are from a structure matrix (i.e., unrotated). Often these are the only loadings and cross-loadings provided by other PLS-based SEM software systems.

As the structure matrix contains the Pearson correlations between indicators and latent variables, this matrix is not particularly meaningful or useful prior to rotation in the context of collinearity or measurement instrument validation. Here the unrotated cross-loadings tend to be fairly high, even when the measurement instrument passes widely used validity and reliability tests. This is generally true for the normalized version of this matrix.

Still, some researchers recommend using the structure loadings and cross-loadings table as well to assess convergent validity, by following two criteria: that the **cross-loadings be lower than 0.5**; and that the **loadings be equal to or greater than 0.5** (Hair et al., 1987; 2009). Note that the loadings here are the same as those provided in the combined loadings and cross-loadings table. The cross-loadings, however, are different. Also, these two criteria generally apply to the version of this table that is *not* normalized.

H.5. View indicator weights

Indicator weights are provided in a table, much in the same way as indicator loadings are (see Figure H.5). All cross-weights are zero, because of the way they are calculated through PLS-based alrgorithms. Each latent variable score is calculated as an exactly linear combination of its indicators, or of its indicators and measurement error, where the weights are multiple regression coefficients linking the indicators to the latent variable.

Figure H.5. Indicator weights window

	Proc	Effi	Effe	ECU	Type (as defined)	SE	P value	VIF	WLS	ES
Proc1	(0.393)	0.000	0.000	0.000	Reflective	0.031	<0.001	1.781	1	0.333
Proc2	(0.421)	0.000	0.000	0.000	Reflective	0.030	<0.001	2.051	1	0.375
Proc3	(0.362)	0.000	0.000	0.000	Reflective	0.035	<0.001	1.593	1	0.292
Effi1	0.000	(0.245)	0.000	0.000	Reflective	0.014	<0.001	3.800	1	0.218
Effi2	0.000	(0.222)	0.000	0.000	Reflective	0.016	<0.001	4.036	1	0.194
Effi3	0.000	(0.238)	0.000	0.000	Reflective	0.019	<0.001	2.414	1	0.194
Effi4	0.000	(0.206)	0.000	0.000	Reflective	0.019	<0.001	2.522	1	0.168
Effi5	0.000	(0.278)	0.000	0.000	Reflective	0.020	<0.001	2.494	1	0.227
Effe1	0.000	0.000	(0.157)	0.000	Reflective	0.013	<0.001	7.843	1	0.147
Effe2	0.000	0.000	(0.152)	0.000	Reflective	0.012	<0.001	9.430	1	0.143
Effe3	0.000	0.000	(0.160)	0.000	Reflective	0.013	<0.001	2.628	1	0.134
Effe4	0.000	0.000	(0.158)	0.000	Reflective	0.012	<0.001	8.708	1	0.150
Effe5	0.000	0.000	(0.149)	0.000	Reflective	0.011	<0.001	6.612	1	0.136
Effe6	0.000	0.000	(0.139)	0.000	Reflective	0.010	<0.001	5.548	1	0.125
Effe7	0.000	0.000	(0.183)	0.000	Reflective	0.013	<0.001	4.209	1	0.165
ECUEmail	0.000	0.000	0.000	(-0.321)	Formative	0.266	0.114	4.434	-1	0.068
ECUEmaillist	0.000	0.000	0.000	(0.405)	Formative	0.297	0.087	4.693	1	0.156

Notes: P values < 0.05 and VIFs < 2.5 are desirable for formative indicators; VIF = indicator variance inflation factor; WLS = indicator weight-loading sign (-1 = Simpson's paradox in l.v.); ES = indicator effect size.

As with indicator loadings, **standard errors are also provided** here for the weights, in the column indicated as "SE", for indicators associated with all latent variables. These standard errors can be used in specialized tests. Among other purposes, they can be used in multi-group analyses, with the same model but different subsamples. Here users may want to compare the measurement models to ascertain equivalence, using a multi-group comparison technique such as the one documented by Kock (2013), and thus ensure that any observed between-group differences in structural model coefficients, particularly in path coefficients, are not due to measurement model differences.

P values are provided for weights associated with all latent variables. These values can also be seen, together with the P values for loadings, as the result of a confirmatory factor analysis. In research reports, users may want to report these P values as an indication that formative latent variable measurement items were properly constructed. This also applies to moderating latent variables that pass criteria for formative measurement, when those variables do not pass criteria for reflective measurement.

As in multiple regression analysis (Miller & Wichern, 1977; Mueller, 1996), it is recommended that **weights with P values that are equal to or lower than 0.05** be considered valid items in a formative latent variable measurement item subset. Formative latent variable indicators whose weights do not satisfy this criterion may be considered for removal.

With these P values, users can also check whether moderating latent variables satisfy validity and reliability criteria for formative measurement, if they do not satisfy criteria for reflective measurement. This can help users demonstrate validity and reliability in hierarchical analyses involving moderating effects, where double, triple etc. moderating effects are tested. For

instance, moderating latent variables can be created, added to the model as standardized indicators, and then their effects modeled as being moderated by other latent variables; an example of double moderation.

In addition to P values, **variance inflation factors (VIFs) are provided** for the indicators of all latent variables, including moderating latent variables. These can be used for indicator redundancy assessment. In reflective latent variables indicators are expected to be redundant. This is not the case with formative latent variables. In formative latent variables indicators are expected to measure *different* facets of the same construct, which means that they should *not* be redundant.

The **VIF threshold of 3.3 has been recommended** in the context of PLS-based SEM in discussions of formative latent variable measurement (Cenfetelli & Bassellier, 2009; Petter et al., 2007). A rule of thumb rooted in the use of this software for many SEM analyses in the past suggests an even more conservative approach: that **capping VIFs to 2.5 for indicators used in formative measurement leads to improved stability** of estimates. The multivariate analysis literature, however, tends to gravitate toward higher thresholds. Also, capping VIFs at 2.5 or 3.3 may in some cases severely limit the number of possible indicators available. Given this, it is recommended that VIFs be capped at 2.5 or 3.3 if this does not lead to a major reduction in the number of indicators available to measure formative latent variables, and if the Cronbach's alpha coefficient associated with the formative latent variable does not fall below 0.6. One example would be the removal of only 2 indicators out of 16 by the use of this rule of thumb, with the Cronbach's alpha coefficient remaining equal to or greater than 0.6. Otherwise, the criteria below should be employed.

Two criteria, one more conservative and one more relaxed, are recommended by the multivariate analysis literature in connection with VIFs; criteria that can arguably also be used in this type of context. More conservatively, **it is recommended that VIFs be lower than 5; a more relaxed criterion is that they be lower than 10** (Hair et al., 1987; 2009; Kline, 1998). High VIFs usually occur for pairs of indicators in formative latent variables, and suggest that the indicators measure the same facet of a formative construct. This calls for the removal of one of the indicators from the set of indicators used for the formative latent variable measurement.

These criteria are generally consistent with formative latent variable theory (see, e.g., Diamantopoulos, 1999; Diamantopoulos & Winklhofer, 2001; Diamantopoulos & Siguaw, 2006). Among other characteristics, formative latent variables are expected, often by design, to have many indicators. Yet, given the nature of multiple regression, indicator weights will normally go down as the number of indicators go up, as long as those indicators are somewhat correlated, and thus P values will normally go up as well. Moreover, as more indicators are used to measure a formative latent variable, the likelihood that one or more will be redundant increases. This will be reflected in high VIFs.

Indicator weight-loading signs (WLS) are provided for the indicators of all latent variables. A negative WLS (i.e., -1) for an indicator means that the indicator in question is making a negative contribution to the R-squared of its latent variable. That is, a negative WLS suggests the existence of a Simpson's paradox instance (Pearl, 2009; Wagner, 1982) in the outer model, associated with a specific indicator assigned to a latent variable. A Simpson's paradox instance in this context is a possible indication of a causality problem, suggesting that a hypothesized link between an indicator and a latent variable is either implausible or reversed. Therefore, **it is recommended that all indicator WLS values be positive**, for both formative and reflective latent variables. Indicators associated with negative WLS values may be considered for removal.

Effect sizes are provided in the column indicated as "ES" for the indicators of all latent variables. As with the effect sizes for paths, the effect sizes for indicators are calculated as the absolute values of the individual contributions of the corresponding indicators to the R-squared coefficients of the latent variable to which each indicator is associated. Similarly to the effect sizes for paths, with the indicator effect sizes users of this software can ascertain whether the indicator effects are small, medium, or large. The values usually recommended are 0.02, 0.15, and 0.35; respectively (Cohen, 1988). Values below 0.02 suggest effects that are too weak to be considered relevant from a practical point of view, even when the corresponding P values are statistically significant. **It is recommended that all indicator effect sizes be equal to or greater than 0.02**, for both formative and reflective latent variables. Indicators with effect sizes that do not meet this criterion may be considered for removal.

H.6. View latent variable coefficients

Several estimates are provided for each latent variable; these can be used in research reports for discussions on the measurement instrument's reliability, discriminant and predictive validity, as well as overall collinearity (see Figure H.6). R-squared, adjusted R-squared, and Q-squared coefficients are provided only for endogenous latent variables; and reflect the percentages of explained variance and predictive validity associated with each of those latent variables, respectively. Composite reliability and Cronbach's alpha coefficients are provided for all latent variables. Also provided for all latent variables are: minimum and maximum values, medians, modes, skewness and excess kurtosis coefficients, results of unimodality and normality tests, and histograms.

Figure H.6. Latent variable coefficients window

	Commor	Easgen	Easund	Complet
R-squared		0.009	0.453	0.003
Adj. R-squared		0.002	0.446	-0.003
Composite reliab.	1.000	0.926	0.897	0.860
Cronbach's alpha	1.000	0.881	0.828	0.755
Avg. var. extrac.	1.000	0.807	0.744	0.672
Full collin. VIF	1.430	1.761	2.291	1.178
Q-squared		0.010	0.454	0.006
Min	-0.997	-2.871	-2.982	-2.329
Max	0.997	1.850	1.931	2.815
Median	0.000	-0.039	-0.034	-0.033
Mode	-0.997	-0.039	-0.034	-0.614
Skewness	-0.000	-0.765	-0.537	0.111
Exc. kurtosis	-2.000	0.561	-0.002	-0.206
Unimodal-RS	No	Yes	Yes	Yes
Unimodal-KMV	No	Yes	Yes	Yes
Normal-JB	No	No	No	Yes
Normal-RJB	No	No	No	Yes
Histogram	View	View	View	View

Composite reliability and Cronbach's alpha coefficients are measures of reliability. Serious questions have been raised regarding Cronbach's alpha's (Cronbach, 1951; Kline, 2010) psychometric properties. However, while the Cronbach's alpha coefficient is reported by this software, and the Factor-Based PLS algorithms employ it as a basis for the estimation of measurement error and composite weights, no assumptions are made about the coefficient's main purported psychometric properties that have been the target of criticism (Sijtsma, 2009). This is an important caveat in light of measurement error theory (Nunnally & Bernstein, 1994). Users should also keep in mind that an alternative and generally more acceptable reliability measure is available, the composite reliability coefficient (Dillon & Goldstein, 1984; Peterson & Yeolib, 2013). Composite reliability coefficients are also known as Dillon–Goldstein rho coefficients (Tenenhaus et al., 2005).

Average variances extracted (AVEs) and full collinearity variance inflation factors (VIFs) are also provided for all latent variables; and are used in the assessment of discriminant validity and overall collinearity, respectively.

Adjusted R-squared coefficients (Theil, 1958; Wooldridge, 1991) are equivalent to R-squared coefficients, with the key difference that they correct for spurious increases in R-squared coefficients due to predictors that add no explanatory value in each latent variable block. Consistently with general recommendations made by Cohen (1988), values of R-squared coefficients and adjusted R-squared coefficients below 0.02 suggest combined effects of predictors in latent variable blocks that are too weak to be considered relevant from a practical point of view. Therefore, **models where R-squared coefficients or adjusted R-squared coefficients are below 0.02 should be considered for revision**, as the explanatory power in sub-models (i.e., latent variable blocks) is below reasonable expectations. Revisions in these models could involve inner and outer model changes, such as removal or change in location of mediating latent variables as well as removal or reassignment of indicators.

The following criteria, one more conservative and the other two more relaxed, are suggested in the assessment of the reliability of a measurement instrument. **These criteria apply only to reflective latent variable indicators**. Reliability is a measure of the quality of a measurement instrument; the instrument itself is typically a set of question-statements. A measurement instrument has good reliability if the question-statements (or other measures) associated with each latent variable are understood in the same way by different respondents.

More conservatively, both the **composite reliability and the Cronbach's alpha coefficients should be equal to or greater than 0.7** (Fornell & Larcker, 1981; Nunnaly, 1978; Nunnally & Bernstein, 1994). The more relaxed version of this criterion, which is widely used, is that one of the two coefficients should be equal to or greater than 0.7. This typically applies to the composite reliability coefficient, which is usually the higher of the two (Fornell & Larcker, 1981). An even more relaxed version sets this threshold at 0.6 (Nunnally & Bernstein, 1994). If a latent variable does not satisfy any of these criteria, the reason will often be one or a few indicators that load weakly on the latent variable. These indicators should be considered for removal.

AVEs are normally used for discriminant validity assessment and, less commonly, for convergent validity assessment. For discriminant validity assessment, AVEs are used in conjunction with latent variable correlations. This is discussed in more detail later, together with the discussion of the table of correlations among latent variables that includes square roots of AVEs. For convergent validity assessment, the AVE threshold frequently recommended for acceptable validity is 0.5 (Fornell & Larcker, 1981), and applies only to reflective latent variables.

Full collinearity VIFs are shown for all latent variables, separately from the VIFs calculated for predictor latent variables in individual latent variable blocks. These VIFs are calculated based on a full collinearity test (Kock & Lynn, 2012), which enables the identification of not only vertical but also lateral collinearity, and allows for a test of collinearity involving all latent variables in a model. Vertical, or classic, collinearity is predictor-predictor latent variable collinearity in individual latent variable blocks. Lateral collinearity is a term coined by Kock & Lynn (2012) that refers to predictor-criterion latent variable collinearity; a type of collinearity that can lead to particularly misleading results. **Full collinearity VIFs can also be used for common method bias tests** (Kock & Lynn, 2012; Lindell & Whitney, 2001) that are more conservative than, and arguably superior to, the traditionally used tests relying on exploratory factor analyses.

A rule of thumb rooted in the use of this software for many SEM analyses in the past suggests that **full collinearity VIFs of 3.3 or lower suggest** the existence of **no multicollinearity** in the model **and no common method bias**. This is also the recommended threshold for VIFs for latent

variables in PLS-based SEM (Kock & Lynn, 2012) and also in slightly different contexts (Cenfetelli & Bassellier, 2009; Petter et al., 2007). On the other hand, two criteria, one more conservative and one more relaxed, are recommended by the multivariate analysis literature in connection with VIFs. They may apply in this type of context as well; although they may be more adequate in path analyses, where all latent variables are measured through single indicators. More conservatively, **it is recommended that VIFs be lower than 5; a more relaxed criterion is that they be lower than 10** (Hair et al., 1987; 2009; Kline, 1998).

Q-squared coefficients are also known as Stone-Geisser Q-squared coefficients, so named after their principal original proponents (Geisser, 1974; Stone, 1974). The Q-squared coefficient is a nonparametric measure traditionally calculated via blindfolding. It is used for the assessment of the predictive validity (or relevance) associated with each latent variable block in the model, through the endogenous latent variable that is the criterion variable in the block. The Q-squared coefficient is sometimes referred to as a resampling analog of the R-squared coefficient. It is often similar in value to that measure; even though the Q-squared coefficient can more easily assume negative values. **Acceptable predictive validity in connection with an endogenous latent variable is suggested by a Q-squared coefficient greater than zero**.

The **unimodality tests** for which results are provided are the Rohatgi- Székely test (Rohatgi & Székely, 1989) and the Klaassen-Mokveld-van Es test (Klaassen et al., 2000). The **normality tests** for which results are provided are the classic Jarque-Bera test (Jarque & Bera, 1980; Bera & Jarque, 1981) and Gel & Gastwirth's (2008) robust modification of this test. Since **these tests are applied to latent variables**, which are combinations either of indicators or of indicators and measurement errors, the outcomes of **these tests can be seen as "multivariate" unimodality and normality** test results.

Both unimodality and normality test results take the form of a "Yes" or "No", meaning that the latent variable distributions are or are not, respectively, unimodal or normal. No unimodality or normality for at least one latent variable (or indicator) is usually seen as an sign that the nonparametric methods used in this software are particularly appropriate. That is, users of this software can justify employing it by noting that not all latent variables are unimodal and normal. It is noteworthy that **the non-normality justification for the use of** non-parametric **PLS-based SEM** methods **has been widely employed** in the past**,** but typically **without any accompanying test of normality!**

H.7. View correlations among latent variables and errors

The "View correlations among latent variables and errors" menu options (see Figure H.7.1) allow users to view tables containing correlations among latent variables, the P values associated with those correlations, square roots of AVEs, correlations among latent variable error terms (or residuals), and the VIFs associated with latent variable error terms (see figures H.7.2 and H.7.3).

Figure H.7.1. Correlations among latent variables and errors options

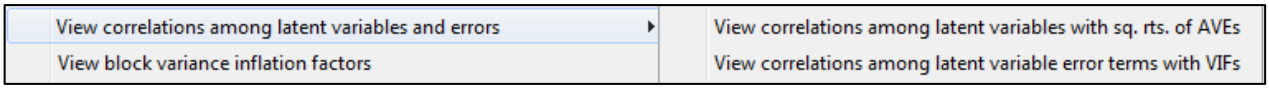

View correlations among latent variables and errors
View block variance inflation factors
View correlations among latent variables with sq. rts. of AVEs
View correlations among latent variable error terms with VIFs

Figure H.7.2. Correlations among latent variables with square roots of AVEs

Correlations among l.vs. with sq. rts. of AVEs

	Proc	Effi	Effe	ECU
Proc	(0.848)	0.460	0.348	0.371
Effi	0.460	(0.842)	0.540	0.101
Effe	0.348	0.540	(0.912)	0.157
ECU	0.371	0.101	0.157	(0.335)

Figure H.7.2. Correlations among latent variable error terms with VIFs

Correlations among l.v. error terms with VIFs

	(e)Proc	(e)Effi	(e)Effe
(e)Proc	(1.001)	0.025	-0.028
(e)Effi	0.025	(1.001)	-0.002
(e)Effe	-0.028	-0.002	(1.001)

In most research reports, users will typically show the table of correlations among latent variables, with the square roots of the average variances extracted on the diagonal, to demonstrate that their measurement instruments pass widely accepted criteria for discriminant validity assessment. A measurement instrument has good discriminant validity if the question-statements (or other measures) associated with each latent variable are not confused by the respondents answering the questionnaire with the question-statements associated with other latent variables, particularly in terms of the meaning of the question-statements.

The following criterion is recommended for discriminant validity assessment: **for each latent variable, the square root of the average variance extracted should be higher than any of the correlations involving that latent variable** (Fornell & Larcker, 1981). That is, the values on the diagonal of the table containing correlations among latent variables, which are the square roots of the average variances extracted for each latent variable, should be higher than any of the values above or below them, in the same column. Or, the values on the diagonal should be higher than any of the values to their left or right, in the same row; which means the same as the previous statement, given the repeated values of the latent variable correlations table.

The above criterion **applies to reflective and formative latent variables, as well as product latent variables representing moderating effects**. If it is not satisfied, the culprit is usually an indicator that loads strongly on more than one latent variable. Also, the problem may involve more than one indicator. You should check the loadings and cross-loadings tables to see if you can identify the offending indicator or indicators, and consider removing them.

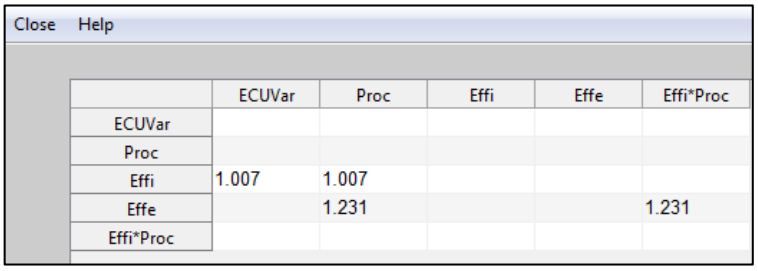
Second to latent variables involved in moderating effects, formative latent variables are the most likely to lead to discriminant validity problems. This is one of the reasons why formative latent variables are not used as often as reflective latent variables in empirical research. In fact, **it is wise to use formative variables sparingly in models** that will serve as the basis for SEM analysis. Formative variables can in many cases be decomposed into reflective latent variables, which themselves can then be added to the model. Often this provides a better understanding of the empirical phenomena under investigation (Edwards, 2011), in addition to helping avoid discriminant validity problems.

A table with **correlations among latent variable error terms containing VIFs associated with the error terms on the diagonal** is also provided. This table may be useful in identifying error terms that are highly correlated, which suggest the existence of confounders. More specifically, if a latent variable A points at a latent variable B, and the error terms (e)A and (e)B are strongly correlated, then this may be an indication of the existence of a hidden confounder. This hidden confounder may be the real cause behind a significant association between A and B, suggesting a causality problem; namely one in which a link may in fact not be a "true" causal link but rather be due to a third variable, the confounder. Particularly problematic are situations in which error terms are so highly correlated that they can be considered redundant, which are indicated by high VIFs in the diagonal of this table. To rule out these situations, and consistently with recommendations by Kock & Lynn (2012), **it is recommended that the VIFs associated with the error terms be equal to or lower than 3.3**.

H.8. View block variance inflation factors

Block variance inflation factors (VIFs) are provided in table format (see Figure H.8) for each latent variable that has two or more predictors in a latent variable block. Block VIFs cannot be calculated for latent variables with only one predictor or no predictor. Here each VIF is associated with one of the two or more predictors, and relates to the link between that predictor and its latent variable criterion. (When one predictor latent variable points at two or more different latent variables in the model, then that latent variable is said to have multiple criteria associated with it.)

Figure H.8. Block variance inflation factors window



In this context, a VIF is a measure of the degree of "vertical" collinearity (Kock & Lynn, 2012), or redundancy, among the latent variables that are hypothesized to affect another latent variable. This classic type of collinearity refers to predictor-predictor collinearity in a latent variable block containing one or more latent variable predictors and one latent variable criterion (Kock & Lynn, 2012). For example, let us assume that there is a block of latent variables in a model, with three latent variables A, B, and C (predictors) pointing at latent variable D. In this case, VIFs are calculated for A, B, and C, and are estimates of the multicollinearity among these predictor latent variables.

A rule of thumb rooted in the use of this software for many SEM analyses in the past, as well as past methodological research, suggests that **block VIFs of 3.3 or lower suggest the existence of no vertical multicollinearity in a latent variable block** (Kock & Lynn, 2012). This is also the recommended threshold for VIFs in slightly different contexts (Cenfetelli & Bassellier, 2009; Petter et al., 2007). On the other hand, two criteria, one more conservative and one more relaxed, are also recommended by the multivariate analysis literature, and can also be seen as applicable in connection with VIFs in this context.

More conservatively, **it is recommended that block VIFs be lower than 5; a more relaxed criterion is that they be lower than 10** (Hair et al., 1987; 2009; Kline, 1998). These criteria may be particularly relevant in the context of **path analyses, where all latent variables are measured through single indicators** (technically, these are not "true" latent variables). The reason why these criteria may be particularly relevant in the context of path analyses is that, without multiple indicators per latent variable, the PLS-based SEM algorithms do not have the "raw material" that they need to reduce collinearity. PLS-based SEM algorithms are particularly effective at reducing collinearity, but chiefly when "true" latent variables are present; that is, when latent variables are measured through multiple indicators.

High block VIFs usually occur for pairs of predictor latent variables, and suggest that the latent variables measure the same construct. If this is not due to indicator assignment problems, it

would arguably call for the removal of one of the latent variables from the block, or from the model.

H.9. View correlations among indicators

The software allows users to view the correlations among all indicators included in the model, in table format. Only the correlations for indicators included in the model are shown through the menu option “View correlations among indicators”, available from the “View and save results” window.

This option is useful for users who want to run a quick check on the correlations among indicators while they are trying to identify possible sources of multicollinearity. This option may also be useful in the identification of candidate indicators for latent variables through the anchor variable procedure developed by Kock & Verville (2012).

The table of correlations among indicators used in the model is usually much larger, with many more columns and rows, than that of the correlations among latent variables. For this reason, the P values for the correlations are not shown in the screen view option together with the correlations, but are saved in the related tab-delimited text file.

To save correlations among all indicators and respective P values, including those indicators not included in the model, use the menu option “Data”, and the appropriate sub-options therein. It should be noted that indicators that are not included in the model are not technically “true” indicators. Nevertheless, they do fall under the general term “manifest variables”; as they are directly measured, and thus not “latent”, variables. They refer to the columns of the original dataset.

The menu option for saving correlations among all manifest variables, which refer to all columns of the original dataset, is available from the main software window under “Data”, after Step 3 is completed. This option is generally more meaningful for users who want to include the correlations among manifest variables in their research reports, as part of a descriptive statistics table, and for users employing the anchor variable procedure developed by Kock & Verville (2012). This option also generates means, standard deviations, and other descriptive statistics for each of the manifest variables. Manifest variables that are not used in the model, and that thus are not “true” indicators, may simply be deleted prior to the inclusion in a research report.

H.10. View/plot linear and nonlinear relationships among latent variables

Choosing the menu option "View/plot linear and nonlinear relationships among latent variables" causes the software to show a table with the types of relationships, warped or linear, between latent variables that are linked in the model (see Figure H.10.1). The term "warped" is used for relationships that are clearly nonlinear, and the term "linear" for linear or quasi-linear relationships. Quasi-linear relationships are slightly nonlinear relationships, which look linear upon visual inspection on plots of the regression curves that best approximate the relationships.

Figure H.10.1. Linear and nonlinear ("warped") relationships among latent variables window

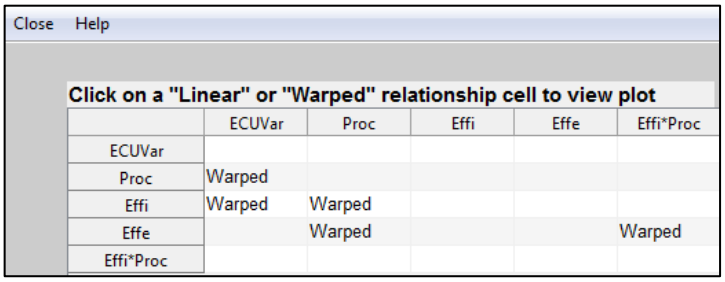
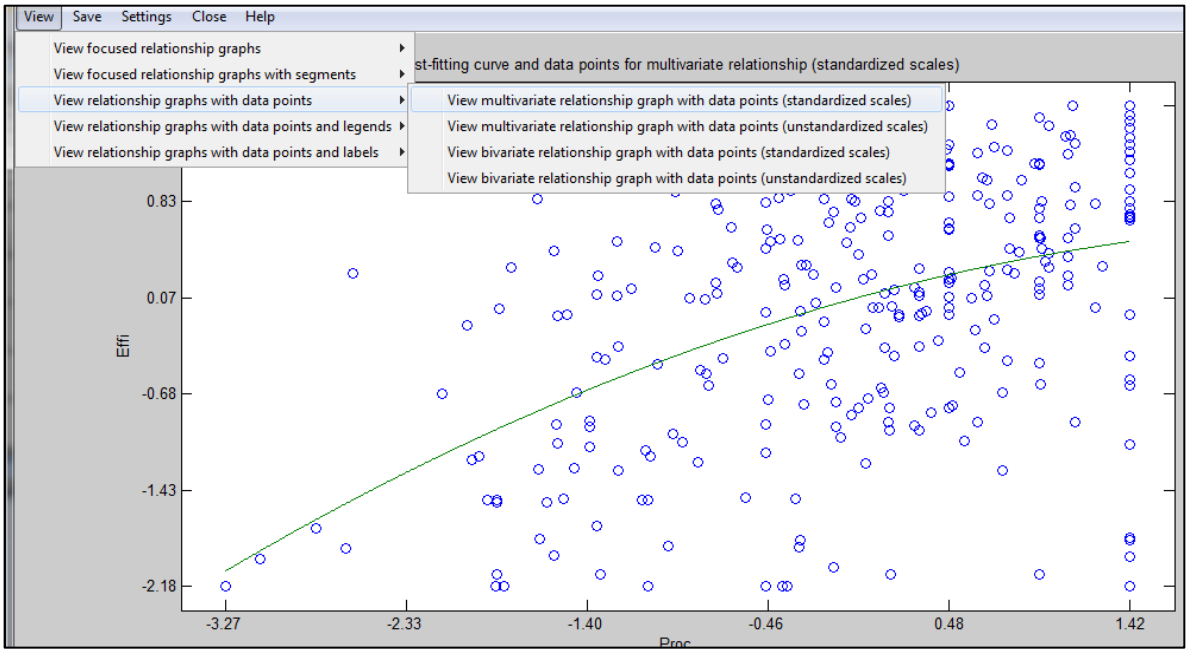


Figure H.10.2. Graph options for direct effects including one with points and best-fitting curve



Several graphs (a.k.a. plots) for direct effects can be viewed by clicking on a cell containing a relationship type description. These cells are the same as those that contain path coefficients, in the path coefficients table that was shown earlier. Among the options available are graphs showing the points as well as the curves that best approximate the relationships (see Figure H.10.2).

The "**View focused relationship graphs**" options allow users to view graphs that focus on the best-fitting line or curve and that exclude data points to provide the effect of zooming in on the best-fitting line or curve area. The options available are: "**View focused multivariate**

relationship graph (standardized scales)", "**View focused multivariate relationship graph (unstandardized scales)**", "**View focused bivariate relationship graph (standardized scales)**", and "**View focused bivariate relationship graph (unstandardized scales)**".

The options above, like other direct effects graph options discussed here, combine variations in terms of two main aspects: whether the scales are **standardized or unstandardized**, and whether the graphs refer to **multivariate or bivariate** relationships.

By default, **latent variable scores are standardized aggregations of indicators**. The latter, namely the indicators, are originally in unstandardized format. Therefore, to obtain the unstandardized equivalents of the latent variable scores, some decisions must be made and extra calculations performed. The unstandardized equivalents of latent variable scores are always approximations.

Unstandardization of scales for latent variable scores, whereby standardized scales are converted to their unstandardized equivalents, is based on the unstandardization option chosen by the user using the "**Settings**" menu option. Three unstandardization options are available: "**Highest loading indicator**", the default option, whereby the mean and standard deviation of the highest loading indicator is used in the unstandardization; "**Average of indicators**", whereby the mean and standard deviation of the average of indicators is used; and "**Weighted average of indicators**", whereby the mean and standard deviation of the weighted average of indicators is used.

Through the "**Settings**" menu option the user can also **set the graph title, the X axis label, and the Y axis label**. The graph title is the text shown at the top of the graph. The X axis label is the text shown next to the X axis, or the horizontal axis. The Y axis label is the text shown next to the Y axis, or the vertical axis.

Multivariate and bivariate relationship graphs usually differ only when two or more predictor latent variables point at one criterion latent variable in a latent variable block. The addition of predictors will normally reduce the path coefficients in a latent variable block. Because of this, **typically a multivariate relationship graph will have a lower overall inclination (or steepness) than its corresponding bivariate relationship graph**. However, this is not always the case. In **statistical suppression instances (MacKinnon et al., 2000), a multivariate relationship graph will have a greater overall inclination than its corresponding bivariate** relationship graph. In **Simpson's paradox instances (Pearl, 2009; Wagner, 1982), multivariate and bivariate relationship graphs will have reversed overall inclinations** – e.g., one will be positive and the other negative.

This software is arguably the first and only, at the time of this writing, to provide both multivariate and bivariate representations of nonlinear relationships. The mathematics underlying the rendering of these representations is complex and somewhat novel. Therefore, these representations should be treated as experimental by users of this software, and any conclusions derived from visual inspection of these representations should be treated with caution.

The "**View focused relationship graphs with segments**" options allow users to view graphs that focus on the best-fitting line or curve, that exclude data points to provide the effect of zooming in on the best-fitting line or curve area, and that show curves as linear segments. The segments are shown with their respective beta coefficients and with or without P values. The options available are: "**View focused multivariate relationship graph with segments (standardized scales)**", "**View focused multivariate relationship graph with segments (standardized scales, P values)**", "**View focused multivariate relationship graph with segments (unstandardized scales)**", "**View focused bivariate relationship graph with**

segments (standardized scales)", "**View focused bivariate relationship graph with segments (standardized scales, P values)**", and "**View focused bivariate relationship graph with segments (unstandardized scales)**".

The number of segments shown in the graphs above depends on the **absolute effect segmentation delta** chosen by the user through the "**Settings**" menu option. This absolute effect segmentation delta is the change (or delta) threshold in the first derivative of the nonlinear function depicting the relationship before a new segment is started. For example, a delta of 0.1 means that in each segment the first derivative of the nonlinear function depicting the relationship does not vary more than 0.1. Since the first derivative does not change in linear relationships, segmentation only occurs in nonlinear relationships. This graph segmentation option allows for the identification of unobserved heterogeneity (Sarstedt & Ringle, 2010) without a corresponding reduction in sample size, providing an alternative to data segmentation approaches such as FIMIX-PLS (Hahn et al., 2002).

The "**View relationship graphs with data points**" options allow users to view graphs with the best-fitting lines or curves and the data points used to produce the best-fitting lines or curves. These options show all the data points, and thus do not provide the effect of zooming in on the best-fitting line or curve area. The options available are: "**View multivariate relationship graph with data points (standardized scales)**", "**View multivariate relationship graph with data points (unstandardized scales)**", "**View bivariate relationship graph with data points (standardized scales)**", and "**View bivariate relationship graph with data points (unstandardized scales)**".

The "**View relationship graphs with data points and legends**" options allow users to view graphs with the best-fitting lines or curves, the data points used to produce the best-fitting lines or curves, and legends associated with data labels. These options show all the data points, and thus do not provide the effect of zooming in on the best-fitting line or curve area. They are useful in cases where many data points are available, because in these cases showing legends instead of data labels next to points avoids graph crowding. The options available are: "**View multivariate relationship graph with data points and legends (standardized scales)**", "**View multivariate relationship graphs with data points and legends (unstandardized scales)**", "**View bivariate relationship graph with data points and legends (standardized scales)**", and "**View bivariate relationship graphs with data points and legends (unstandardized scales)**".

The "**View relationship graphs with data points and labels**" options allow users to view graphs with the best-fitting lines or curves, the data points used to produce the best-fitting lines or curves, and data labels next to the data points to which they refer. These options show all the data points, and thus do not provide the effect of zooming in on the best-fitting line or curve area. They are useful in cases where few data points are available, because in these cases showing data labels next to points provides a clear picture of what each data point refers to without graph crowding. The options available are: "**View multivariate relationship graph with data points and labels (standardized scales)**", "**View multivariate relationship graphs with data points and labels (unstandardized scales)**", "**View bivariate relationship graph with data points and labels (standardized scales)**", and "**View bivariate relationship graphs with data points and labels (unstandardized scales)**".

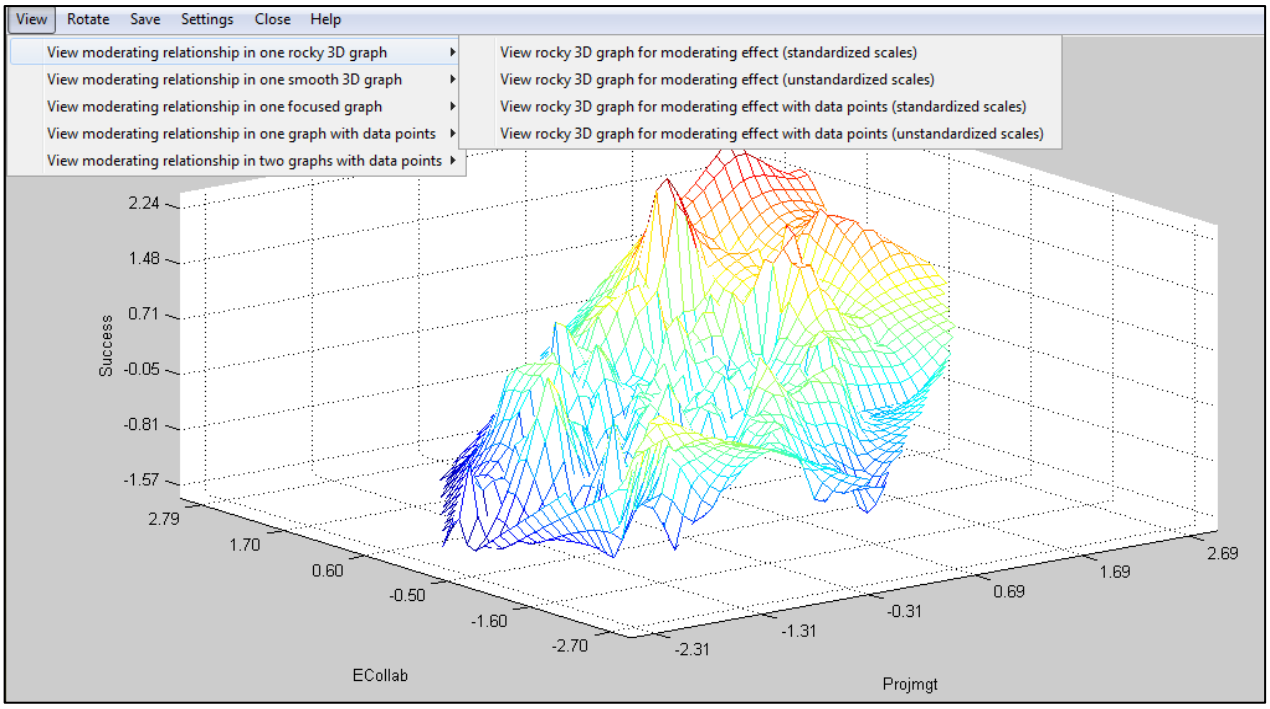
As mentioned earlier in this manual, the Warp2 and the Warp2 Basic algorithms try to identify a U-curve relationship between each pair of predictor-criterion latent variables, and, if that relationship exists, the algorithm used transforms (or "warps") the scores of the predictor latent variables so as to better reflect the U-curve relationship in the estimated path coefficients in the

model. The Warp3 and the Warp3 Basic algorithms, the former being the default algorithm used by this software, try to identify a relationship defined by a function whose first derivative is a U-curve. This type of relationship follows a pattern that is more similar to an S-curve (or a somewhat distorted S-curve), and can be seen as a combination of two connected U-curves, one of which is inverted.

Sometimes a Warp3-based analysis will lead to results that tell you that a relationship between two latent variables has the form of a U-curve or a line, as opposed to an S-curve. Similarly, sometimes a Warp2-based analysis will tell you that a relationship has the form of a line. This is because the underlying algorithms find the type of relationship that best fits the distribution of points associated with a pair of latent variables, and sometimes those types are not S-curves or U-curves.

As with direct effects, several graphs (a.k.a. plots) for moderating effects can be viewed by clicking on a cell containing a relationship type description. These cells are the same as those that contain path coefficients, in the path coefficients table that was shown earlier. Their column labels are displayed on the table as product latent variables (e.g., Effi*Proc). In this example, namely Effi*Proc, the latent variable Effi is hypothesized to moderate the relationship between Proc and another latent variable, where Proc points at the third latent variable. The third latent variable is listed in the corresponding row label. Among the options available are 3-dimensional (3D) graphs showing the points as well as the surfaces that best approximate the relationships (see Figure H.10.3).

Figure H.10.3. Graph options for moderating effects including 3D graph with points and best-fitting surface



Moderating relationships involve three latent variables, the moderating variable and the pair of variables that are connected through a direct link. The **sign and strength of a path coefficient for a moderating relationship** refer to the **effect of the moderating variable on the sign and strength of the path for the direct relationship** that it moderates. For example, if the path for

the direct relationship has its sign going from negative to positive and becomes significantly stronger in that direction as one moves from the low to the high range of the moderating variable, then the sign of the path coefficient for the corresponding moderating relationship will be positive and the path coefficient will be relatively high; possibly high enough to yield a statistically significant effect.

No moderating relationship graph currently available from this software accurately represents the true nature of a moderating relationship. This comment seems to apply to all other publicly available SEM software tools; to the best of our knowledge, and at the time of this writing. Therefore various graphs are provided so that users can choose the one that in their view best illustrates the relationship. **An accurate representation of a moderating relationship would be that of a multivariate distortion** in the surface representing the relationship. **The distortion refers to a "twisting" of the surface around the moderating variable axis**, with a multivariate adjustment, and with corresponding changes in the overall inclinations of the sections of the surface representing the direct effect being moderated. The mathematical underpinnings of such representation were still under development at the time of this writing, and may be available for implementation in future versions of this software.

The "**View moderating relationship in one rocky 3D graph**" options allow users to view 3D graphs where the surfaces are generated through Delaunay triangulations (Chew, 1989; Lee & Schachter, 1980) without smoothing. Surfaces can be viewed with data points excluded or included. The displays with data points excluded are analogous to those used in the focused 2D graphs. The options to view surfaces with data points excluded are: "**View rocky 3D graph for moderating effect (standardized scales)**" and "**View rocky 3D graph for moderating effect (unstandardized scales)**". The options to view surfaces with data points included are: "**View rocky 3D graph for moderating effect with data points (standardized scales)**", and "**View rocky 3D graph for moderating effect with data points (unstandardized scales)**".

The "**Rotate**" menu option allows the user to rotate a 3D graph up, down, left, and right. Through the "**Settings**" menu option the user can **set the following 3D graph options: the graph title, the moderating variable (M) axis label, the X axis label, and the Y axis label**. The graph title is the text shown at the top of the graph. The M axis label is the text shown next to the moderating variable axis. The X axis label is the text shown next to the X axis, or the predictor variable axis. The Y axis label is the text shown next to the Y axis, or the criterion variable axis.

The "**View moderating relationship in one smooth 3D graph with data points**" options allow users to view 3D graphs where the surfaces are generated through Delaunay triangulations (Chew, 1989; Lee & Schachter, 1980) with smoothing. Because the surfaces are generated *with* smoothing, they sometimes resemble more bed sheets than rocky mountain formations. Surfaces can be viewed with data points excluded or included. The options to view surfaces with data points excluded are: "**View smooth 3D graph for moderating effect (standardized scales)**" and "**View smooth 3D graph for moderating effect (unstandardized scales)**". The options to view surfaces with data points included are: "**View smooth 3D graph for moderating effect with data points (standardized scales)**", and "**View smooth 3D graph for moderating effect with data points (unstandardized scales)**".

In addition to 3D graphs, this software also provides various 2-dimensional (2D) graphs of moderating relationships. The 2D graphs shown for moderating relationships refer to low and high values of the moderating variable, and display the relationships of the variables connected through the moderated direct links in those ranges.

The "**View moderating relationship in one focused graph**" options allow users to view 2D moderating effect graphs that focus on the best-fitting lines or curves for high and low values of the moderating variable, and that exclude data points to provide the effect of zooming in on the area comprising the best-fitting lines or curves. The options available are: "**View focused graph with low-high values of moderating variable (standardized scales)**", and "**View focused graph with low-high values of moderating variable (unstandardized scales)**".

Through the "**Settings**" menu option the user can also **set the following 2D moderating effect graph options: the graph title, the labels associated with high and low values of the moderating variable, and the location of the legend box containing these labels**. These options allow users to create more informative 2D moderating relationship graphs. For example, instead of "Low Exp" and "High Exp", more informative labels such as "Novices" and "Veterans" could be used. Setting the location of the legend box (e.g., from "East" to "Northwest") allows users to move the legend box from more to less crowded areas of the graph, giving the graph a more balanced and "cleaner" appearance.

The "**View moderating relationship in one graph with data points**" options allow users to view 2D moderating effect graphs with the best-fitting lines or curves for high and low values of the moderating variable, and the data points used to produce the best-fitting lines or curves. These options show all the data points, and thus do not provide the effect of zooming in on the area comprising the best-fitting lines or curves. The options available are: "**View graph with low-high values of moderating variable and data points (standardized scales)**", and "**View graph with low-high values of moderating variable and data points (unstandardized scales)**".

The "**View moderating relationship in two graphs with data points**" options allow users to view 2D moderating effect graphs with the best-fitting lines or curves for high and low values of the moderating variable, and the data points used to produce the best-fitting lines or curves, in two graphs shown side-by-side. These options show all the data points, and thus do not provide the effect of zooming in on the areas comprising the best-fitting lines or curves. The options available are: "**View two graphs with low-high values of moderating variable and data points (standardized scales)**", and "**View two graphs with low-high values of moderating variable and data points (unstandardized scales)**".

The graphs of relationships between pairs of latent variables, and between latent variables and links (moderating relationships), provide a much more nuanced view of how latent variables are related. However, **caution must be taken in the interpretation of these graphs, especially when the distribution of data points is very uneven**.

An extreme example would be a warped graph in which all of the data points would be concentrated on the right part of the graph, with only one data point on the far left part of the graph. That single data point, called an outlier, could strongly influence the shape of the nonlinear relationship. In cases such as this, the researcher must decide whether the outlier is "good" data that should be allowed to shape the relationship, or is simply "bad" data resulting from a data collection error.

If the outlier is found to be "bad" data, it can be removed from the analysis, even as it remains in the dataset, by a simple procedure. The user should first add the latent variable score to the set of standardized indicators used in a SEM analysis, using the appropriate menu option under the option "Modify", from the main software window, after Step 5 is completed. The user can then remove the outlier by restricting the values assumed by the latent variable, using the appropriate selections under the "Settings" options, to a range that excludes the outlier. This allows for the

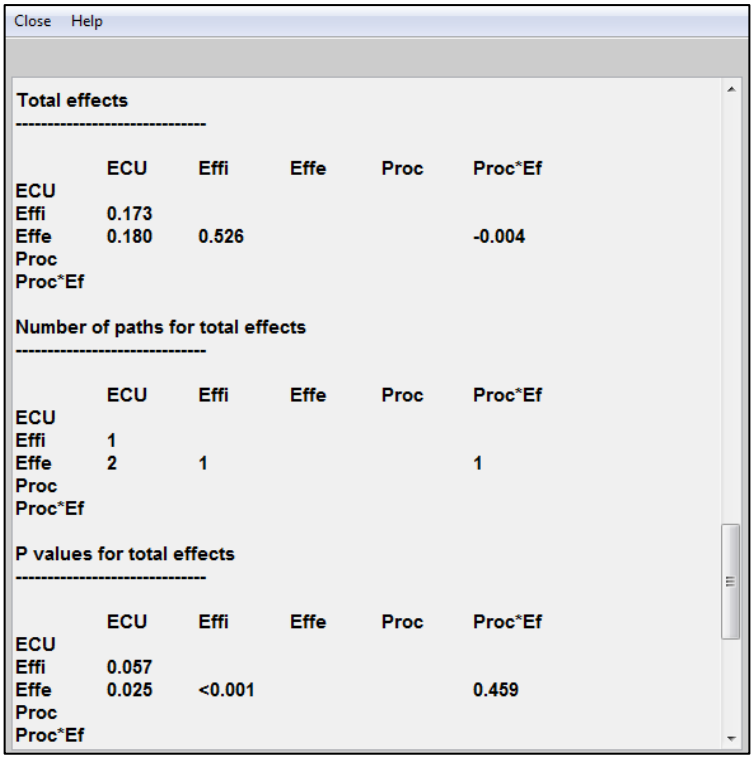
exclusion of the outlier without the user having to modify and re-read a dataset. This procedure may lead to a visible change in the shape of the nonlinear relationship, and significantly affect the results.

An outlier that is found to be “bad” data can also be removed from the dataset, and thus from the analysis, by a more time-consuming procedure. The user should first save the latent variable scores into a file, using the appropriate Save” menu option in the results window, after Step 5 is completed. Then the user should add those scores to the original dataset; the rows will be in the same order. Next the user should open the modified dataset with a spreadsheet software tool (e.g., Excel). The outlier should be easy to identify on the dataset (e.g., a value greater than 4), and should be eliminated. Then the user should re-read this modified file as if it was the original data file, and run all of the SEM analysis steps again.

H.11. View indirect and total effects

Through the "**View indirect and total effects**" options the software allows users to view outputs for indirect and total effects (Bollen, 1987; Kock & Gaskins, 2014) associated with all latent variables that are linked via one or more paths with more than one segment. The options available are "**View indirect and total effects (table view)**" and "**View indirect and total effects (classic view)**". The difference between these two options is that the former shows indirect and total effects outputs in extendable table format, and the latter in wrapped text format. The former option, corresponding to the table view, is recommended in complex models with many links among latent variables. Figure H.11 illustrates the latter option, the classic view, which is so named because it was the option used in previous versions of the software.

Figure H.11. Indirect and total effects window



For each set of indirect and total effects, the following values are provided: the path coefficients associated with the effects, the number of paths that make up the effects, the P values associated with effects (calculated via resampling, using the selected resampling method), the standard errors associated with the effects, and effect sizes associated with the effects.

Indirect effects are aggregated for paths with a certain number of segments. As such, the software provides separate reports, within the same output window, for paths with 2, 3 etc. segments. The software also provides a separate report for sums of indirect effects, as well as for total effects. All of these reports include P values, standard errors, and effect sizes.

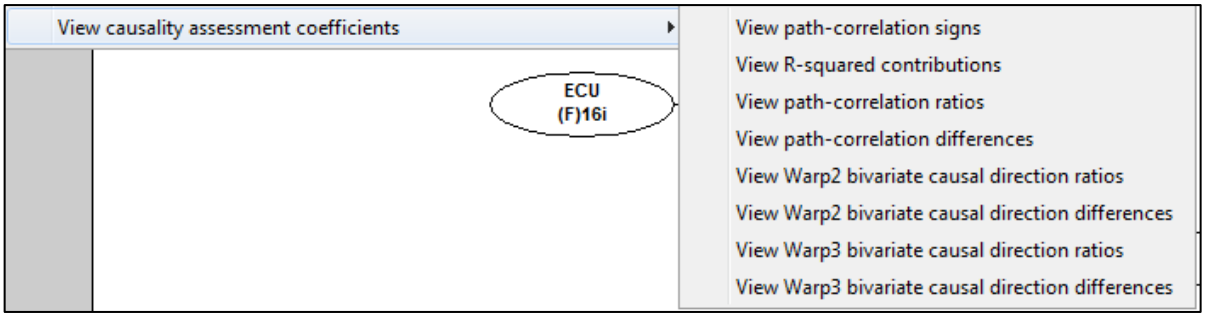
Having access to indirect and total effects can be critical in the evaluation of downstream effects of latent variables that are mediated by other latent variables, especially in complex models with multiple mediating effects along concurrent paths. Indirect effects also allow for direct estimations, via resampling, of the P values associated with mediating effects that have

traditionally relied on non-automated and thus time-consuming calculations based on linear (Preacher & Hayes, 2004) and nonlinear (Hayes & Preacher, 2010) assumptions.

H.12. View causality assessment coefficients

The "**View causality assessment coefficients**" options allow users to view a number of coefficients associated with individual paths that can be used in causality assessment. The options available are: "**View path-correlation signs**", "**View R-squared contributions**", "**View path-correlation ratios**", "**View path-correlation differences**", "**View Warp2 bivariate causal direction ratios**", "**View Warp2 bivariate causal direction differences**", "**View Warp3 bivariate causal direction ratios**", and "**View Warp3 bivariate causal direction differences**". The topic of causality assessment in the context of SEM is controversial (Pearl, 2009). Therefore, these causality assessment coefficients should be treated as experimental by users of this software, and any conclusions derived from them should be treated with caution.

Figure H.12: Causality assessment coefficients options



The "**View path-correlation signs**" option allows users to identify path-specific Simpson's paradox instances (Pearl, 2009; Wagner, 1982), by inspecting a table with the path-correlation signs (shown in the table as the values 1 and -1). A negative path-correlation sign, or the value -1, is indicative of a Simpson's paradox instance. A Simpson's paradox instance is a possible indication of a causality problem, suggesting that a hypothesized path is either implausible or reversed.

The **interpretation of individual Simpson's paradox instances can be difficult**. This may be especially the case with demographic variables when these are included in the model as control variables, suggesting what may appear to be unlikely or impossible reverse directions of causality. For example, let us say that a negative path-correlation sign occurs when we include the control variable "Age" (time from birth, measured in years) into a model pointing at the variable "Job performance" (self-assessed, measured through multiple indicators on Likert-type scales). This may be interpreted as suggesting that "Job performance" causes "Age" in the sense that increased job performance causes someone to age, or causes time to pass faster.

Alternative explanations frequently exist for Simpson's paradox instances, as well as for other "red flags" suggested by causality assessment coefficients. Taking the example above, one possible alternative explanation is that increased job performance causes employment to be maintained at more advanced ages, supporting the direction of causality from "Job performance" to "Age" instead of the reverse path. It can also mean that, because of sampling problems, those with greater job performance included in the sample tended to be older. Yet another alternative explanation is that there is no link between "Job performance" and "Age", and that the inclusion of another control variable artificially induces that link; which tends to happen when path coefficients are associated with negligible R-squared contributions (i.e., lower than 0.02). Whatever the case may be, ideally models should be free from Simpson's paradox instances,

because, as noted below, these instances generally detract from the explanatory power of the model.

Because an instance of Simpson's paradox occurs when a path coefficient and a correlation associated with a pair of linked variables have different signs, the corresponding contribution to the R-squared of the criterion variable in the latent variable block where it occurs is negative (see Mueller, 1996; for a discussion of this effect in the context of evolutionary biology, see Kock, 2011). The "**View R-squared contributions**" option allows users to view the values of the individual contributions to the R-squared of the criterion variable in each latent variable block by each of the predictor latent variables in the block.

The "**View path-correlation ratios**" option allows users to identify statistical suppression instances (MacKinnon et al., 2000), by inspecting a table with the absolute path-correlation ratios. These ratios are calculated by dividing path coefficients by their respective correlation coefficients and taking the absolute values of those divisions. An instance of statistical suppression occurs when a path coefficient is greater, in absolute terms, than the corresponding correlation associated with a pair of linked variables. This leads to a path-correlation ratio that is greater than 1. Like a Simpson's paradox instance, a statistical suppression instance is a possible indication of a causality problem (Spirtes et al., 1993), suggesting that a hypothesized path is either implausible or reversed. The following interpretations are suggested for absolute path-correlation ratios: ratio > 1 indicates statistical suppression; 1 < ratio <= 1.3: weak suppression; 1.3 < ratio <= 1.7: medium; 1.7 < ratio: strong.

In the same way that one can distinguish between a statistically significant and non-significant direct association, one can also distinguish between statistically significant and non-significant suppression instances. The "**View path-correlation differences**" option allows users to do just that, by inspecting a table with the absolute path-correlation differences and their respective P values. The absolute path-correlation differences can be used together with the absolute path-correlation ratios to identify paths that need special attention, because path-correlation ratios alone can sometimes provide an inflated perception of problems, especially when paths and correlations are both very small. Generally speaking, a path that meets the following criteria should be seen as referring to a link that needs special attention in terms of possible elimination or careful interpretation: **absolute path-correlation ratio greater than 1.3, and P value for absolute path-correlation difference equal to a lower than 0.05**. However, a path that meets these criteria will not necessarily be associated with causality problems; it may in fact suggest a particularly interesting and unique finding (see, e.g., MacKinnon et al., 2000).

One useful and interesting property of nonlinear algorithms, such as the Warp2 and Warp3 algorithms, is that often bivariate nonlinear coefficients of association calculated using those algorithms vary depending on the hypothesized direction of causality. That is, they tend to be stronger in one direction than the other, which means that the residual (or error) is greater when the hypothesized direction of causality is in one way or the other. As such, they can be used, together with other coefficients, as partial evidence in support or against hypothesized causal links.

The "**View Warp2 bivariate causal direction ratios**" option allows users to identify instances in which the Warp2 algorithm suggests that causality may be reversed, by inspecting a table with the Warp2 bivariate causal direction ratios. These ratios are calculated by dividing the path coefficient obtained for the reversed link by the path coefficient obtained for the link with the hypothesized direction. The following interpretations are suggested for Warp2 bivariate

causal direction ratios: ratio > 1 supports reversed link; 1 < ratio <= 1.3: weak support; 1.3 < ratio <= 1.7: medium; 1.7 < ratio: strong.

In the same way that one can distinguish between a statistically significant and non-significant direct association, one can also distinguish between a statistically significant and non-significant Warp2 bivariate causal direction reversal instance. The "**View Warp2 bivariate causal direction differences**" option allows users to do just that, by inspecting a table with the absolute Warp2 bivariate causal direction differences and their respective P values. The absolute Warp2 bivariate causal direction differences can be used together with the Warp2 bivariate causal direction ratios to identify paths that need special attention, because Warp2 bivariate causal direction ratios alone can sometimes provide an inflated perception of problems, especially when paths in one direction and the other are both very small. Generally speaking, a path that meets the following criteria should be seen as referring to a link that needs special attention in terms of possible direction reversal: **Warp2 bivariate causal direction ratio greater than 1.3, and P value for absolute Warp2 bivariate causal direction difference equal to a lower than 0.05**.

Since the Warp3 and Warp2 algorithms are different, a similar set of outputs exists in connection with Warp3 bivariate causal direction inferences to those available for Warp2. Often these different sets outputs will be fairly consistent, but sometimes they will not.

The "**View Warp3 bivariate causal direction ratios**" option allows users to identify instances in which the Warp3 algorithm suggests that causality may be reversed, by inspecting a table with the Warp3 bivariate causal direction ratios. These ratios are calculated by dividing the path coefficient obtained for the reversed link by the path coefficient obtained for the link with the hypothesized direction. The following interpretations are suggested for Warp3 bivariate causal direction ratios: ratio > 1 supports reversed link; 1 < ratio <= 1.3: weak support; 1.3 < ratio <= 1.7: medium; 1.7 < ratio: strong.

The "**View Warp3 bivariate causal direction differences**" option allows users to distinguish between a statistically significant and non-significant Warp3 bivariate causal direction reversal instance. Users can do that by inspecting a table with the absolute Warp3 bivariate causal direction differences and their respective P values. The absolute Warp3 bivariate causal direction differences can be used together with the Warp3 bivariate causal direction ratios to identify paths that need special attention, because Warp3 bivariate causal direction ratios alone can sometimes provide an inflated perception of problems, especially when paths in one direction and the other are both very small. Generally speaking, a path that meets the following criteria should be seen as referring to a link that needs special attention in terms of possible direction reversal: **Warp3 bivariate causal direction ratio greater than 1.3, and P value for absolute Warp3 bivariate causal direction difference equal to a lower than 0.05**.

Since ratios and P values are generated for the Warp2 and Warp3 algorithms, a more relaxed approach would be to consider for special attention in terms of possible direction reversal only links that meet both the criteria for Warp2 and Warp3 above. Another approach, also somewhat relaxed, would be to consider for special attention only links that meet the criteria that refer to the nonlinear algorithm used for the calculation of the path coefficient associated with the link, either Warp2 or Warp3 (this includes the "basic" options). Having said that, the above criteria arguably apply to paths calculated using the Linear algorithm.

The extent to which using more or less relaxed approaches would lead to "false positives" and "false negatives" in terms of support and lack of support for hypothesized directions of causality is an issue that will require future research, particularly research employing Monte Carlo simulations (Robert & Casella, 2010) where the true directions of causality are known.

A path meeting the above criteria for both Warp2 and Warp3 algorithms, in terms of support for causal direction reversal, may in some cases appear to lead to an absurd conclusion. Upon further consideration, however, reversing the path may not sound as absurd. For example, let us say that a path from the variable "Age" (time from birth, measured in years) pointing at the variable "Job performance" (self-assessed, measured through multiple indicators on Likert-type scales) meets the above criteria, suggesting that it should be reversed. This may be interpreted as suggesting that "Job performance" causes "Age" in the sense that increased job performance causes someone to age, or causes time to pass faster. These could be seen as absurd conclusions, even if we consider work as a possible cause of oxidative stress, and thus accelerated decrepitude (note that "Age" is defined as time from birth, measured in years). However, a different interpretation is that increased job performance causes employment to be maintained at more advanced ages, supporting the direction of causality from "Job performance" to "Age" in a more reasonable and intuitively appealing way.

I. Concluding remarks and additional issues

This software provides users with a wide range of features, including experimental features and also other features that are not available from other SEM software. For example, this software is the first and only (at the time of this writing) to explicitly identify nonlinear functions connecting pairs of latent variables in SEM models and calculate coefficients of association accordingly.

A wide range of features means that there are many coefficients, graphs and other elements that users can choose to include in research reports, and many possible interpretations of those elements. This user manual does not cover all possible interpretations. Users are strongly advised to keep abreast of the latest developments on methodological issues employing this software, particularly those from research published in academic outlets (e.g., academic journals).

Multivariate statistical analysis software systems, like this software, are inherently complex; sometimes yielding results that are biased and disconnected with the reality of the phenomena being modeled. Users are strongly cautioned against accepting the results provided by this software as a completely unbiased representation of the underlying reality that the software attempts to unveil.

No multivariate statistical analysis software yields completely unbiased results. If one such "perfect" software tool existed, the percentages of false positives and false negatives based on a number of trials with the software using simulated data would all be zero. That is, no false positives or false negatives of any kind (e.g., association strength, direction of causality) would occur.

Achieving this level of perfection is the driving force behind the development of this software, even though this level of perfection will never be achieved – simply because it is not achievable. Some additional issues regarding this quest for perfection are discussed in the following subsections.

I.1. Warping from a conceptual perspective

What this software does when it "warps" relationships is relatively simple at a conceptual level. It identifies a set of functions F1(LVp1), F2(LVp2) … that relate blocks of latent variable predictors (LVp1, LVp2 ...) to a criterion latent variable (LVc) in this way:

$$LVc = p1*F1(LVp1) + p2*F2(LVp2) + \ldots + E$$

In the equation above, p1, p2 ... are path coefficients, and E is the error term of the equation. All variables are standardized. Any model can be decomposed into a set of blocks relating latent variable predictors and criteria in this way.

Typically, the more the functions F1(LVp1), F2(LVp2) ... look like curves, and unlike lines, the greater is the difference between the path coefficients p1, p2 ... and those that would have been obtained through a strictly linear analysis.

What this software does is not unlike what a researcher would do if he or she modified predictor latent variable scores prior to the calculation of path coefficients using a function like the logarithmic function. An example is provided in the equation below, where a logarithmic transformation is applied to LVp1.

$$LVc = p1*\log(LVp1) + p2*LVp2 + \ldots + E$$

This software, however, does that automatically and for a much wider range of functions, with modification constants included. For example, in the term A*log(B*LVp1) the constants A and B are modification constants; using simply log(LVp1) as a modifier function in an equation like the one above assumes that A=1 and B=1, which may be incorrect assumptions that will lead to distorted results and mistaken conclusions.

As mentioned above, often the path coefficients p1, p2 ... will go up in value due to warped analysis, but that may not always be the case. Given the nature of multivariate analysis, an increase in a path coefficient may lead to a decrease in a different path coefficient, for predictor latent variables associated with the same criterion latent variable, because each path coefficient in a block is calculated in a way that controls for the effects of the other predictor latent variables. That is, in any given block of latent variables, the predictor latent variables "compete" for the explained variance in the criterion latent variable.

There is no guarantee that the functions F1(LVp1), F2(LVp2) ... discovered by this software will match perfectly that "true" underlying functions. As mentioned earlier, this level of perfection is one that should be strived for, but that is essentially impossible to achieve due to one key factor – measurement error.

The more measurement error exists (i.e., the greater is its magnitude), the more likely it is that the functions F1(LVp1), F2(LVp2) ... discovered by this software will be distorted by error. In fact, the existence of significant measurement error may lead this software to model relationships that are actually linear as nonlinear.

With the above caveats in mind, users can check, through simple visual inspection tests, whether the functions discovered by this software are at least good approximations of the true underlying functions.

To do so, users can divide the dataset into a number of quantiles (e.g., 3), and then build graphs containing the mean values of each criteria latent variable for each of the quantiles. These graphs can be simple bar charts or scatter plots.

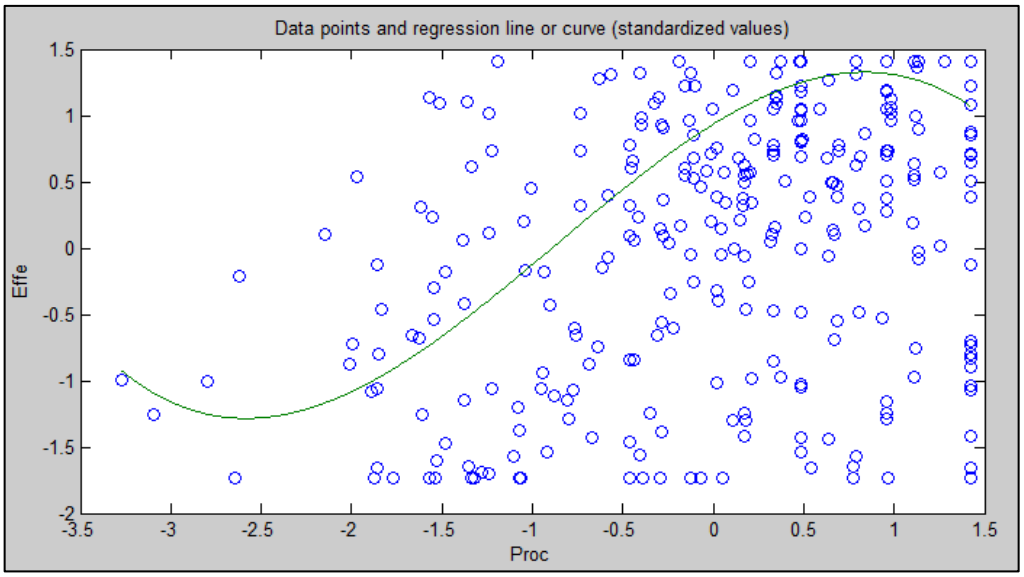
This allows users to check whether the shapes of the plots are similar to the shapes of the best-fitting curves generated by this software. The best-fitting curves are representations of the functions F1(LVp1), F2(LVp2) ... discovered by this software.

I.2. Interpreting warped relationships

Linear relationships between pairs of latent variables, that is, those relationships best described by a line, are relatively easy to interpret. They suggest that an increase in one variable either leads to an increase (if the slope of the line is positive) or decrease (if the slope is negative) in the other variable.

Nonlinear relationships provide a much more nuanced view of the data, but at the same time are much more difficult to interpret. Figure I.2 shows what could be seen as a distorted S curve that is fitted to the data points. The latent variables are "Proc", the extent to which various teams charged with developing new products kept track of their work and costs (i.e., engaged in procedural structuring); and "Effe", the effectiveness of the teams, measured as the market success in terms of sales and profits of the new products that the teams developed.

Figure I.2. Example of warped relationship



The distorted S can in turn be seen as a combination of two distorted U curves (or J curves), one straight and the other inverted, connected at an inflection point. The inflection point is the point at the curve where the curvature changes direction; i.e., the second derivative of the S curve changes sign. The inflection point is located at around minus 1 standard deviations from the "Proc" mean. That mean is at the zero mark on the horizontal axis, since the data shown is standardized.

Because an S curve is a combination of two distorted U curves, we can interpret each U curve section separately. A straight U curve, like the one shown on the left side of the graph, before the inflection point, can be interpreted as follows.

The first half of the U curve goes from approximately minus 3.4 to minus 2.5 standard deviations from the mean, at which point the lowest team effectiveness value is reached for the U curve. In that first half of the U curve, an increase in team procedural structuring leads to a decrease in team effectiveness. After that first half, an increase in team procedural structuring leads to an increase in team effectiveness.

One interpretation is that the first half of the U curve refers to novice users of procedural structuring techniques. That is, the process of novice users struggling to use procedural

structuring techniques more and more intensely, which they may not be familiar with, ends up leading to effectiveness losses for their teams. At a certain point, around minus 2.5 standard deviations, that situation changes, and the teams start to really benefit from procedural structuring, possibly because the second half of the U curve refers to users with more experience using procedural structuring techniques.

The interpretation of the second U curve on the right, this one an inverted U curve, should be done in a similar fashion. Usually there are multiple interpretations that can be plausible depending on context and other data. Other data may include qualitative data, which can be very useful when combined with quantitative data.

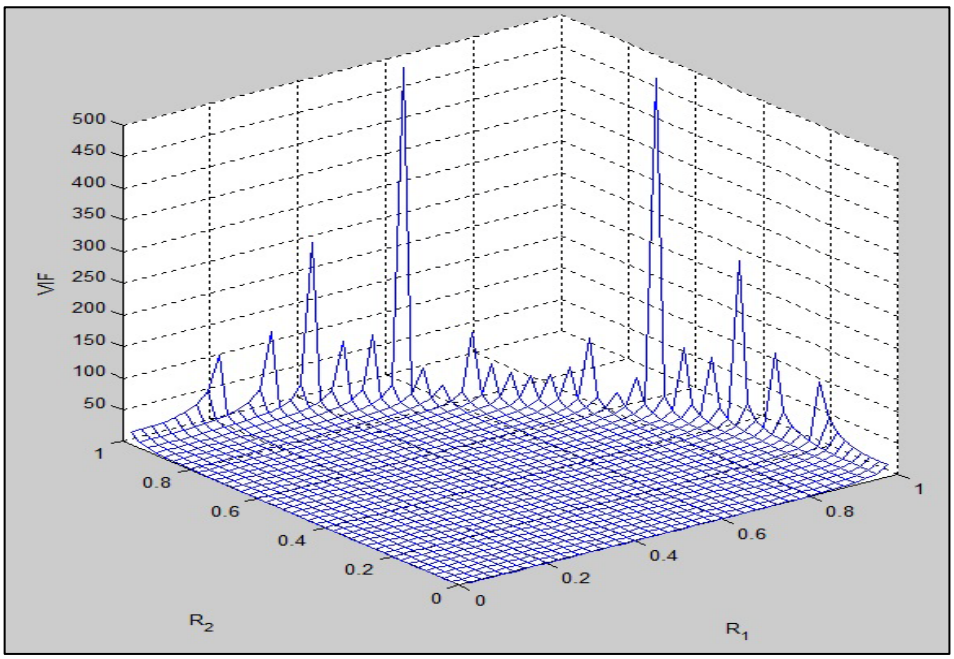
As can be inferred from this example, it is not easy to interpret nonlinear relationships. But the apparent simplicity of strictly linear modeling, or linear estimations of possibly nonlinear relationships, is nothing but a mirage.

I.3. Correlation versus collinearity

Let us consider a theoretical case in which two predictor variables point at a criterion variable, and the predictor variables are uncorrelated. In this case, the value of the R for the criterion variable (the positive square root of the R-squared) will be a function of two other correlation values, R_1 and R_2, which are the correlations between each of the predictor variables and the criterion. The value of the VIF, which is itself a function of R, will consequently be a function of R_1 and R_2.

The values of the VIF for the scenario above are plotted in Figure I.3, generated based on a simulation with MATLAB. Three dimensions are needed because three variables are involved. As it can be seen, the variable VIF can reach unacceptably high values, clearly suggestive of collinearity, and for much lower values of R_1 and R_2 than in the case when only two variables are present. Let us assume that we were to set the threshold of VIF for collinearity at 3.3. In this case, a correlation of 0.835 or higher would suggest collinearity in a situation involving only two variables (Kock & Lynn, 2012).

Figure I.3. The relationship between the VIF and the Rs for three variables



The points at which the VIF values increase steeply are indicated as peaks (including small peaks) on the three-dimensional plot. Here a combination of values of R_1 and R_2 in the range of 0.6 to 0.8 lead to VIF values that are suggestive of collinearity for a threshold level of 3.3. For example, if R_1 and R_2 are both equal to 0.625, the corresponding VIF will be 4.57.

As models become more complex from a structural perspective, with more variables in them, the absolute values of the correlations that can lead to significant multicollinearity goes progressively down. Even if not in the same block, latent variables may still be redundant and cause interpretation problems when correlations are relatively low. This is why it is important that users of this software take the various VIFs that are reported into consideration when assessing their models.

The example above also illustrates the fact that the concepts of collinearity and correlation are distinct concepts, even though they are often confused. Collinearity is a multivariate notion, whereas correlation refers to a pair of variables (Kock & Lynn, 2012). Two or more variables are said to be collinear when they measure the same attribute of an object; the latter is also called a construct. In this sense, the variables "satisfaction with a technology" and "excitement about the technology" may be collinear, if the question-statements related to these two variables are seen as referring to the same object attribute "affective response to the technology" by the respondents of a questionnaire. Two variables are said to be correlated if they vary in concert with each other, even though the variables may measure totally different object attributes; e.g., a person's weight from 1 to 20 years of age, and the price of gasoline during those years.

I.4. Stable P value calculation methods

A Monte Carlo simulation was conducted to assess the performance of three P value calculation methods implemented through this software: Bootstrapping, Stable2, and Stable3. Performance was assessed in terms of statistical power and closeness to the actual standard errors obtained through the analyses of simulated samples. Standard errors are used, together with path coefficients, to obtain P values. Table I.4, adapted from Kock (2014b), summarizes the results of this simulation.

Table I.4. Summarized Monte Carlo experiment results for P value calculation methods

Method	**BOOT**	**STBL2**	**STBL3**	**BOOT**	**STBL2**	**STBL3**
Sample size	**50**	**50**	**50**	**300**	**300**	**300**
CO>GT(TruePath)	0.450	0.450	0.450	0.450	0.450	0.450
CO>GT(AvgPath)	0.383	0.383	0.383	0.388	0.388	0.388
CO>GT(Power)	0.905	0.954	0.946	1	1	1
CO>GT(SEPath)	0.125	0.125	0.125	0.076	0.076	0.076
CO>GT(EstSEPath)	0.120	0.115	0.122	0.047	0.053	0.054
CO>EU(TruePath)	0.400	0.400	0.400	0.400	0.400	0.400
CO>EU(AvgPath)	0.347	0.347	0.347	0.347	0.347	0.347
CO>EU(Power)	0.781	0.900	0.867	1	1	1
CO>EU(SEPath)	0.131	0.131	0.131	0.072	0.072	0.072
CO>EU(EstSEPath)	0.133	0.116	0.124	0.049	0.053	0.055
CO>AC(TruePath)	0.250	0.250	0.250	0.250	0.250	0.250
CO>AC(AvgPath)	0.224	0.224	0.224	0.218	0.218	0.218
CO>AC(Power)	0.419	0.611	0.559	0.985	0.995	0.994
CO>AC(SEPath)	0.141	0.141	0.141	0.061	0.061	0.061
CO>AC(EstSEPath)	0.166	0.118	0.129	0.054	0.054	0.056
GT>SU(TruePath)	0.500	0.500	0.500	0.500	0.500	0.500
GT>SU(AvgPath)	0.333	0.333	0.333	0.347	0.347	0.347
GT>SU(Power)	0.711	0.863	0.823	1	1	1
GT>SU(SEPath)	0.206	0.206	0.206	0.160	0.160	0.160
GT>SU(EstSEPath)	0.146	0.116	0.125	0.052	0.053	0.055
EU>SU(TruePath)	0.230	0.230	0.230	0.230	0.230	0.230
EU>SU(AvgPath)	0.175	0.175	0.175	0.163	0.163	0.163
EU>SU(Power)	0.254	0.410	0.356	0.917	0.921	0.906
EU>SU(SEPath)	0.131	0.131	0.131	0.085	0.085	0.085
EU>SU(EstSEPath)	0.157	0.119	0.132	0.054	0.054	0.056
AC>SU(TruePath)	0.200	0.200	0.200	0.200	0.200	0.200
AC>SU(AvgPath)	0.159	0.159	0.159	0.147	0.147	0.147
AC>SU(Power)	0.240	0.405	0.335	0.866	0.868	0.849
AC>SU(SEPath)	0.137	0.137	0.137	0.073	0.073	0.073
AC>SU(EstSEPath)	0.165	0.119	0.132	0.053	0.054	0.056

The column labels BOOT, STBL2 and STBL3 respectively refer to the Bootstrapping, Stable2, and Stable3 methods. The latent variables in the model used as a basis for the simulation are: CO = communication flow orientation; GT = usefulness in the development of IT solutions; EU = ease of understanding; AC = accuracy; and SU = impact on redesign success (for more details, see Kock, 2014b). The meanings of the acronyms within parentheses are the following: TruePath = true path coefficient; AvgPath = mean path coefficient estimate; Power = statistical power; SEPath = standard error of path coefficient estimate; and EstSEPath = method-specific standard error of path coefficient estimate.

To conduct the simulation we created an analyzed 1,000 samples for each of the following sample sizes: 50, 100, 200, 300, and 500. The PLS Mode A algorithm was used in the analyses. In this summarized set of results we restrict ourselves to sample sizes 50 and 300. Full results, for all sample sizes included in the simulation, are available from Kock (2014b).

As we can see, the mean path coefficient estimates differ from the true path coefficients across different sample sizes, and generally underestimate the true path coefficients. This underestimation stems from the use of composites in PLS Mode A, which in turn leads to the known composite *correlation* attenuation (Nunnally & Bernstein, 1994). This attenuation "propagates" to the path coefficients (Kock, 2014). This problem is addressed in this software through the availability of Factor-Based PLS algorithms.

Generally the method-specific standard errors of path coefficient estimates obtained via Stable3 were the closest to the actual (or true) standard errors of path coefficient estimates. This suggests that standard errors estimated via Stable3 are not only stable when compared with those estimated via Bootstrapping, but also more accurate. Moreover, both Stable2 and Stable3 led to greater statistical power than Bootstrapping at small sample sizes. This is noteworthy, because power tends to be compromised the most with small sample sizes, and to invariably increase as sample sizes go up regardless of the standard error and P value calculation method used.

I.5. Missing data imputation methods

A Monte Carlo simulation was conducted to assess the performance of five missing data imputation methods implemented through this software: Arithmetic Mean Imputation, Multiple Regression Imputation, Hierarchical Regression Imputation, Stochastic Multiple Regression Imputation, and Stochastic Hierarchical Regression Imputation. Table I.5, adapted from Kock (2014c), summarizes the results of this simulation.

Table I.5. Summarized Monte Carlo experiment results for missing data imputation methods

Missing data imputation scheme	NMD	MEAN	MREGR	HREGR	MSREG	HSREG
CO>GT(TruePath)	0.450	0.450	0.450	0.450	0.450	0.450
CO>GT(AvgPath)	0.390	0.348	0.367	0.354	0.333	0.300
CO>GT(SEPath)	0.075	0.113	0.110	0.113	0.138	0.162
CO>EU(TruePath)	0.400	0.400	0.400	0.400	0.400	0.400
CO>EU(AvgPath)	0.349	0.312	0.321	0.313	0.289	0.262
CO>EU(SEPath)	0.069	0.101	0.108	0.106	0.133	0.151
CO>AC(TruePath)	0.250	0.250	0.250	0.250	0.250	0.250
CO>AC(AvgPath)	0.219	0.198	0.206	0.195	0.188	0.161
CO>AC(SEPath)	0.062	0.078	0.090	0.083	0.100	0.108
GT>SU(TruePath)	0.500	0.500	0.500	0.500	0.500	0.500
GT>SU(AvgPath)	0.381	0.357	0.359	0.352	0.334	0.312
GT>SU(SEPath)	0.127	0.152	0.156	0.158	0.179	0.195
EU>SU(TruePath)	0.230	0.230	0.230	0.230	0.230	0.230
EU>SU(AvgPath)	0.192	0.183	0.199	0.178	0.188	0.163
EU>SU(SEPath)	0.062	0.072	0.077	0.078	0.082	0.089
AC>SU(TruePath)	0.200	0.200	0.200	0.200	0.200	0.200
AC>SU(AvgPath)	0.165	0.157	0.176	0.154	0.166	0.141
AC>SU(SEPath)	0.058	0.067	0.073	0.072	0.077	0.081
GT3<GT(TrueLoad)	0.700	0.700	0.700	0.700	0.700	0.700
GT3<GT(AvgLoad)	0.811	0.691	0.606	0.649	0.623	0.652
GT3<GT(SELoad)	0.113	0.042	0.120	0.076	0.115	0.090

The column labels NMD, MEAN, MREGR, HREGR, MSREG and HSREG respectively refer to no missing data, Arithmetic Mean Imputation, Multiple Regression Imputation, Hierarchical Regression Imputation, Stochastic Multiple Regression Imputation, and Stochastic Hierarchical Regression Imputation. The latent variables in the model used as a basis for the simulation are: CO = communication flow orientation; GT = usefulness in the development of IT solutions; EU = ease of understanding; AC = accuracy; and SU = impact on redesign success (for more details, see Kock, 2014c). The meanings of the acronyms within parentheses are the following: TruePath = true path coefficient; AvgPath = mean path coefficient estimate; SEPath = standard error of path coefficient estimate; TrueLoad = true loading; AvgLoad = mean loading estimate; and SELoad = standard error of loading estimate.

When creating data for our Monte Carlo simulation we varied the following conditions: percentage of missing data (0%, 30%, 40%, and 50%), and sample size (100, 300, and 500). This led to a 4 x 3 factorial design, with 12 conditions. We created an analyzed 1,000 samples for each of these 12 conditions; a total of 12,000 samples. In this summarized set of results we restrict ourselves to 30% missing data and the sample size of 300. Full results, for all percentages of missing data and sample sizes included in the simulation, are available from Kock (2014c). Since all loadings are the same in the true population model, loading-related estimates for only

one indicator of the composites are shown. This avoids crowding and repetition, as the same pattern of results repeats itself in connection with all loadings.

The mean path coefficient estimates that are shown underlined were obtained through the application of the PLS Mode A algorithm to datasets where no data was missing (NMD). Note that they generally underestimate the true path coefficients. This underestimation stems from the use of composites, discussed earlier, which leads to an attenuation of composite *correlations* (Nunnally & Bernstein, 1994). This correlation attenuation extends to the path coefficients (Kock, 2014), leading to the observed underestimation. The opposite effect is observed in connection with loadings, which tend to be overestimated in PLS-based SEM analyses employing PLS Mode A. As noted earlier, these problems are addressed in this software through the availability of Factor-Based PLS algorithms.

Multiple Regression Imputation (MREGR) yielded the least biased mean path coefficient estimates, followed by Arithmetic Mean Imputation (MEAN). When we look at mean loading estimates, Arithmetic Mean Imputation (MEAN) yielded the least biased results, followed by Stochastic Hierarchical Regression Imputation (HSREG) and Hierarchical Regression Imputation (HREGR.

Compared with the no missing data condition (NMD), none of the methods induced a reduction in standard errors for path coefficients. This is noteworthy since prior results outside the context of PLS-based SEM have tended to show a significant downward bias in standard errors, particularly for non-stochastic missing data imputation varieties. Such downward bias in standard errors has led to concerns regarding an inflation in type I errors, and warnings against the use of single missing data imputation methods in general (Enders, 2010; Newman, 2014). Our results suggest that such concerns may not be warranted in the context of PLS-based SEM.

I.6. Factor-Based PLS algorithms

The Factor-Based PLS algorithms available in this software combine the precision of covariance-based SEM algorithms, under common factor model assumptions (Kock, 2014), with the nonparametric characteristics of classic PLS algorithms. Moreover, the Factor-Based PLS algorithms address head-on a problem that has been discussed since the 1920s – the factor indeterminacy problem. Classic PLS algorithms yield composites, as linear combinations of indicators, which can be seen as factor approximations. The Factor-Based PLS algorithms, on the other hand, provide estimates of the true factors, as linear combinations of indicators and measurement errors.

A Monte Carlo simulation was conducted to comparatively assess the performance of one of the Factor-Based PLS algorithms, namely the Factor-Based PLS Type CFM1 algorithm, against that of the PLS Mode A algorithm. Like covariance-based SEM algorithms, the Factor-Based PLS Type CFM1 algorithm is fully compatible with common factor model assumptions, including the assumption that all indicator errors are uncorrelated. Table I.6, adapted from Kock (2014), summarizes the results of this simulation.

Table I.6. Summarized Monte Carlo experiment results for composite-based and factor-based algorithms

SEM method	PLSA	PLSF	PLSA	PLSF	PLSA	PLSF
Sample size	50	50	100	100	300	300
EU>TE(TruePath)	0.400	0.400	0.400	0.400	0.400	0.400
EU>TE(AvgPath)	0.339	0.380	0.309	0.385	0.303	0.394
EU>TE(SEPath)	0.125	0.161	0.128	0.127	0.110	0.070
EU>TP(TruePath)	0.300	0.300	0.300	0.300	0.300	0.300
EU>TP(AvgPath)	0.260	0.301	0.248	0.294	0.234	0.297
EU>TP(SEPath)	0.135	0.157	0.108	0.133	0.085	0.079
TE>TP(TruePath)	0.200	0.200	0.200	0.200	0.200	0.200
TE>TP(AvgPath)	0.201	0.234	0.189	0.225	0.174	0.203
TE>TP(SEPath)	0.144	0.163	0.098	0.132	0.061	0.079
EU3<EU(TrueLoad)	0.700	0.700	0.700	0.700	0.700	0.700
EU3<EU(AvgLoad)	0.793	0.692	0.802	0.695	0.808	0.699
EU3<EU(SELoad)	0.129	0.108	0.113	0.077	0.112	0.049

The column labels PLSA and PLSF respectively refer to the PLS Mode A and Factor-Based PLS Type CFM1 algorithms. The latent variables in the model used as a basis for the simulation are: EU = e-collaboration technology use; TE = team efficiency; and TP = team performance (for more details, see Kock, 2014). The meanings of the acronyms within parentheses are the following: TruePath = true path coefficient; AvgPath = mean path coefficient estimate; SEPath = standard error of path coefficient estimate; TrueLoad = true loading; AvgLoad = mean loading estimate; and SELoad = standard error of loading estimate.

In the Monte Carlo simulation 300 samples were created for each of the following sample sizes: 50, 100, and 300. We show results for all of the structural paths in the model, but restrict ourselves to loadings for one indicator in one factor since all loadings are the same in the true population model used. This is also done to avoid repetition, as the same general pattern of results for loadings repeats itself for all indicators in all factors.

As we can see from the summarized results, the Factor-Based PLS Type CFM1 algorithm yielded virtually unbiased estimates at the sample size of 300, whereas the PLS Mode A algorithm yielded significantly biased estimates at that same sample size. One of the reasons for

these significantly biased estimates with PLS Mode A are the relatively low loadings in the true population model used as a basis for simulation, namely 0.7 for all indicators, which tend to be a challenge for algorithms based on Wold's original PLS design (Kock, 2014).

The relatively low loadings in the true population model apparently had little effect on the Factor-Based PLS Type CFM1 algorithm's asymptotic convergence to the true values of the model parameters, although those loadings probably slowed down that convergence somewhat as sample sizes increased. In other simulations we conducted with higher loadings, convergence was achieved at smaller sample sizes.

For several of the path coefficients and loadings the Factor-Based PLS Type CFM1 algorithm yielded lower standard errors, particularly as sample sizes increased. This is noteworthy because the Factor-Based PLS Type CFM1 algorithm is considerably more computationally complex than the PLS Mode A algorithm (Kock, 2014), and thus could have been expected to have a greater "cost" in terms of standard errors.

Nevertheless, standard errors yielded at the sample size of 50 were generally higher for the Factor-Based PLS Type CFM1 algorithm. Apparently the difference was enough to have a negative effect on power, as the ratios of path coefficients to standard errors indicate. That is, at the sample size of 50 one could argue based on the results that the PLS Mode A algorithm has greater power than the Factor-Based PLS Type CFM1 algorithm for this particular model, although the ratios of path coefficients to standard errors suggest that both algorithms may struggle to avoid type II errors at this small sample size, particularly for the paths whose true coefficients were lower than 0.400 (the path with the highest strength).

J. Glossary

Adjusted R-squared coefficient. A measure equivalent to the R-squared coefficient, with the key difference that it corrects for spurious increases in the R-squared coefficient due to predictors that add no explanatory value in each latent variable block. Like R-squared coefficients, adjusted R-squared coefficients can assume negative values. These are rare occurrences that normally suggest problems with the model in which they occur; e.g., severe collinearity or model misspecification.

Average variance extracted (AVE). A measure associated with a latent variable, which is used in the assessment of the discriminant validity of a measurement instrument. Less commonly, it can also be used for convergent validity assessment.

Composite reliability coefficient. This is a measure of reliability associated with a latent variable. Another name for it is Dillon–Goldstein rho coefficient. Unlike the Cronbach's alpha coefficient, another measure of reliability, the compositive reliability coefficient takes indicator loadings into consideration in its calculation. It often is slightly higher than the Cronbach's alpha coefficient.

Construct. A conceptual entity measured through a latent variable. Sometimes it is referred to as "latent construct". The terms "construct" or "latent construct" are often used interchangeably with the term "latent variable".

Convergent validity of a measurement instrument. Convergent validity is a measure of the quality of a measurement instrument; the instrument itself is typically a set of question-statements. A measurement instrument has good convergent validity if the question-statements (or other measures) associated with each latent variable are understood by the respondents in the same way as they were intended by the designers of the question-statements.

Cronbach's alpha coefficient. This is a measure of reliability associated a latent variable. It usually increases with the number of indicators used, and is often slightly lower than the composite reliability coefficient, another measure of reliability.

Discriminant validity of a measurement instrument. Discriminant validity is a measure of the quality of a measurement instrument; the instrument itself is typically a set of question-statements. A measurement instrument has good discriminant validity if the question-statements (or other measures) associated with each latent variable are not confused by the respondents, in terms of their meaning, with the question-statements associated with other latent variables.

Endogenous latent variable. This is a latent variable that is hypothesized to be affected by one or more other latent variables. An endogenous latent variable has one or more arrows pointing at it in the model graph.

Exogenous latent variable. This is a latent variable that does not depend on other latent variables, from a SEM analysis perspective. An exogenous latent variable does not have any arrow pointing at it in the model graph.

Factor score. A factor score is the same as a latent variable score; see the latter for a definition.

Formative latent variable. A formative latent variable is one in which the indicators are expected to measure certain attributes of the latent variable, but the indicators are not expected to be highly correlated with the latent variable score, because they (i.e., the indicators) are not expected to be correlated with one another. For example, let us assume that the latent variable "Satisf" ("satisfaction with a meal") is measured using the two following question-statements: "I am satisfied with the main course" and "I am satisfied with the dessert". Here, the meal

comprises the main course, say, filet mignon; and a dessert, a fruit salad. Both main course and dessert make up the meal (i.e., they are part of the same meal) but their satisfaction indicators are not expected to be highly correlated with each other. The reason is that some people may like the main course very much, and not like the dessert. Conversely, other people may be vegetarians and hate the main course, but may like the dessert very much.

Indicator. The term indicator is frequently used as synonymous with that of manifest variable; a convention that is used here. Thus, see the latter for a definition. More technically though, indicators are manifest variables that are actually used in the measurement model as direct measures of latent variables. As such, technically speaking, there can be manifest variables that are not indicators, if the manifest variables in question are part of the original dataset but not included in the measurement model.

Inner model. In a structural equation modeling analysis, the inner model is the part of the model that describes the relationships among the latent variables that make up the model. In this sense, the path coefficients are inner model parameter estimates.

Latent variable. A latent variable is a variable that is measured through multiple variables called indicators or manifest variables. For example, "satisfaction with a meal" may be a latent variable measured through two manifest variables that store the answers on a 1 to 7 scale (1=strongly disagree; 7 strongly agree) to the following question-statements: "I am satisfied with this meal", and "After this meal, I feel full".

Latent variable block. A latent variable block is a group of latent variables in which one or more predictor latent variables point at one criterion latent variable. In a PLS-based SEM analysis, once latent variable scores are calculated, a series of multiple least squares regressions are conducted to calculate path coefficients. Each multiple least squares regression is performed on a latent variable block, until all blocks are covered. The term "latent variable block" is also used in the PLS-based SEM literature to refer to a group of manifest variables linked to their assigned latent variable; i.e., a latent variable and its indicators.

Latent variable score. Latent variable scores are values calculated based on the indicators defined by the user as associated with the latent variable. They are calculated using one of the outer model analysis algorithms available. These scores may be understood as new columns in the data, with the same number of rows as the original data (unless a range-restricted analysis is conducted), and which generally tend to maximize the loadings and minimize the cross-loadings of a pattern matrix of loadings after an oblique rotation.

Latent variable error. An error variable that accounts for the variance in an endogenous latent variable that is not accounted for by the latent variable predictors that point at the endogenous latent variable. The terms "error" and "residual" are used interchangeably in this document. Nevertheless, they refer to subtly different entities. Technically speaking, the term "error" typically refers to the error variable in the true population model, which is assumed to be uncorrelated with latent variables other than the endogenous latent variable to which it is associated. Conversely, the term "residual" typically refers to the corresponding estimated error, the difference between the expected value of the latent variable and its point estimate, which in practice is often correlated with latent variables other than the endogenous latent variable to which it is associated. This is an example of a broader occurrence in multivariate analyses: more often than not sample-specific estimates violate assumptions about the theoretical true values, even if slightly.

Manifest variable. A manifest variable is one of several variables that are used to indirectly measure a latent variable. For example, "satisfaction with a meal" may be a latent variable

measured through two manifest variables, which assume as values the answers on a 1 to 7 scale (1=strongly disagree; 7 strongly agree) to the following question-statements: “I am satisfied with this meal”, and “After this meal, I feel full”.

Outer model. In a SEM analysis, the outer model is the part of the model that describes the relationships among the latent variables that make up the model and their indicators. In this sense, the weights and loadings are outer model parameter estimates.

Portable document format (PDF). This is an open standard file format created by Adobe Systems, and widely used for exchanging documents. It is the format used for this software’s documentation.

Q-squared coefficient. This measure is also known after its main proponents as the Stone-Geisser Q-squared coefficient (Geisser, 1974; Stone, 1974). The Q-squared coefficient is a nonparametric measure traditionally calculated via blindfolding. It is used for the assessment of the predictive validity (or relevance) associated with each latent variable block in the model, through the endogenous latent variable that is the criterion variable in the block. The Q-squared coefficient is sometimes referred to as a resampling analog of the R-squared. It is often similar in value to that measure. The Q-squared coefficient can assume negative values.

Reflective latent variable. A reflective latent variable is one in which all of the indicators are expected to be highly correlated with the latent variable score, and also highly correlated with one another. For example, the answers to certain question-statements by a group of people, measured on a 1 to 7 scale (1=strongly disagree; 7 strongly agree) and answered after a meal, are expected to be highly correlated with the latent variable “satisfaction with a meal”. The question-statements are: “I am satisfied with this meal”, and “After this meal, I feel full”. Therefore, the latent variable “satisfaction with a meal”, can be said to be reflectively measured through these two indicators. These indicators store answers to the two question-statements. This latent variable could be represented in a model graph as “Satisf”, and the indicators as “Satisf1” and “Satisf2”.

Reliability of a measurement instrument. Reliability is a measure of the quality of a measurement instrument; the instrument itself is typically a set of question-statements. A measurement instrument has good reliability if the question-statements (or other measures) associated with each latent variable are understood in the same way by different respondents.

R-squared coefficient. This is a measure calculated only for endogenous latent variables, and that reflects the percentage of explained variance for each of those latent variables. The higher the R-squared coefficient, the better is the explanatory power of the predictors of the latent variable in the model, especially if the number of predictors is small. Contrary to popular belief and in spite of what their name implies, R-squared coefficients are not calculated by squaring a correlation-like measure. They can assume negative values, although these are rare occurrences that normally suggest problems with the model in which they occur; e.g., severe collinearity or model misspecification.

Structural equation modeling (SEM). A general term used to refer to a class of multivariate statistical methods where complex relationships among latent variables and indicators are estimated at once. In a SEM analysis, each latent variable is typically measured through multiple indicators, although there may be cases in which only one indicator is used to measure a latent variable. Key measures of relationships among latent variables are path coefficients (or standardized partial regression coefficients) and corresponding P values. Key measures of relationships among latent variables and their respective indicators are weights and loadings, and corresponding P values.

Variance inflation factor (VIF). This is a measure of the degree of collinearity (or multicollinearity) among variables, including both indicators and latent variables. With latent variables, collinearity can take two main forms: vertical and lateral collinearity (Kock & Lynn, 2012). Vertical, or classic, collinearity is predictor-predictor latent variable collinearity in individual latent variable blocks. Lateral collinearity is a term coined by Kock & Lynn (2012) that refers to predictor-criterion latent variable collinearity; a type of collinearity that can lead to particularly misleading results. Full collinearity VIFs allow for the simultaneous assessment of both vertical and lateral collinearity in a SEM model.

K. Acknowledgements

The author would like to thank the users of WarpPLS for their questions, comments, and suggestions. New features are frequently added in response to requests by users. Revised text and other materials from previously published documents by the author have been used in the development of this manual.

L. References

Adelman, I., & Lohmoller, J.-B. (1994). Institutions and development in the nineteenth century: A latent variable regression model. *Structural Change and Economic Dynamics*, 5(2), 329-359.

Baron, R. M., & Kenny, D. A. (1986). The moderator–mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. *Journal of Personality & Social Psychology*, 51(6), 1173-1182.

Bera, A.K., & Jarque, C.M. (1981). Efficient tests for normality, homoscedasticity and serial independence of regression residuals: Monte Carlo evidence. *Economics Letters*, 7(4), 313-318.

Berglund, E., Lytsy, P., & Westerling, R. (2012). Adherence to and beliefs in lipid-lowering medical treatments: A structural equation modeling approach including the necessity-concern framework. *Patient Education and Counseling*, 91(1), 105-112.

Biong, H., & Ulvnes, A.M. (2011). If the supplier's human capital walks away, where would the customer go? *Journal of Business-to-Business Marketing*, 18(3), 223-252.

Bollen, K.A. (1987). Total, direct, and indirect effects in structural equation models. *Sociological Methodology*, 17(1), 37-69.

Brewer, T.D., Cinner, J.E., Fisher, R., Green, A., & Wilson, S.K. (2012). Market access, population density, and socioeconomic development explain diversity and functional group biomass of coral reef fish assemblages. *Global Environmental Change*, 22(2), 399-406.

Cenfetelli, R., & Bassellier, G. (2009). Interpretation of formative measurement in information systems research. *MIS Quarterly*, 33(4), 689-708.

Chatelin, Y.M., Vinzi, V.E., & Tenenhaus, M. (2002), *State-of-art on PLS path modeling through the available software*. Storrs, CT: Department of Economics, University of Connecticut.

Chew, L.P. (1989). Constrained Delaunay triangulations. *Algorithmica*, 4(1-4), 97-108.

Chin, W.W., Marcolin, B.L., & Newsted, P.R. (2003). A partial least squares latent variable modeling approach for measuring interaction effects: Results from a Monte Carlo simulation study and an electronic-mail emotion/adoption study. *Information Systems Research*, 14(2), 189-218.

Chiquoine, B., & Hjalmarsson, E. (2009). Jackknifing stock return predictions. *Journal of Empirical Finance*, 16(5), 793-803.

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences*. Hillsdale, NJ: Lawrence Erlbaum.

Cronbach, L.J. (1951). Coefficient alpha and the internal structure of tests. *Psychometrika*, 16(3), 297–334.

Dempster, A.P., Laird, N.M., & Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39 (1), 1-38.

Diamantopoulos, A. (1999). Export performance measurement: Reflective versus formative indicators. *International Marketing Review*, 16(6), 444-457.

Diamantopoulos, A., & Siguaw, J.A. (2006). Formative versus reflective indicators in organizational measure development: A comparison and empirical illustration. *British Journal of Management*, 17(4), 263–282.

Diamantopoulos, A., & Winklhofer, H. (2001). Index construction with formative indicators: An alternative scale development. *Journal of Marketing Research*, 37(1), 269-177.

Dillon, W.R., & Goldstein, M. (1984). *Multivariate analysis: Methods and applications*. New York, NY: Wiley.

Edwards, J.R. (2011). The fallacy of formative measurement. *Organizational Research Methods*, 14(2), 370-388.

Efron, B., Rogosa, D., & Tibshirani, R. (2004). Resampling methods of estimation. In N.J. Smelser, & P.B. Baltes (Eds.). *International Encyclopedia of the Social & Behavioral Sciences (pp. 13216-13220)*. New York, NY: Elsevier.

Ehremberg, A.S.C., & Goodhart, G.J. (1976). *Factor analysis: Limitations and alternatives*. Cambridge, MA: Marketing Science Institute.

Enders, C.K. (2010). *Applied missing data analysis*. New York, NY: Guilford Press.

Ferguson, G.A. (1981). *Statistical analysis in psychology and education*. New York, NY: McGraw-Hill.

Fornell, C., & Larcker, D.F. (1981). Evaluating structural equation models with unobservable variables and measurement error. *Journal of Marketing Research*, 18(1), 39-50.

Geisser, S. (1974). A predictive approach to the random effects model. *Biometrika*, 61(1), 101-107.

Gel, Y.R., & Gastwirth, J.L. (2008). A robust modification of the Jarque–Bera test of normality. *Economics Letters*, 99(1), 30-32.

Giaquinta, M. (2009). *Mathematical analysis: An introduction to functions of several variables*. New York, NY: Springer.

Goodhue, D.L., Lewis, W., & Thompson, R. (2012). Does PLS have advantages for small sample size or non-normal data? *MIS Quarterly*, 36(3), 981-1001.

Guo, K.H., Yuan, Y., Archer, N.P., & Connelly, C.E. (2011). Understanding nonmalicious security violations in the workplace: A composite behavior model. *Journal of Management Information Systems*, 28(2), 203-236.

Hahn, C., Johnson, M.D., Herrmann, A., & Huber, F. (2002). Capturing customer heterogeneity using a finite mixture PLS approach. *Schmalenbach Business Review*, 54(3), 243-269.

Hair, J.F., Anderson, R.E., & Tatham, R.L. (1987). *Multivariate data analysis*. New York, NY: Macmillan.

Hair, J.F., Black, W.C., Babin, B.J., & Anderson, R.E. (2009). *Multivariate data analysis*. Upper Saddle River, NJ: Prentice Hall.

Hayes, A. F., & Preacher, K. J. (2010). Quantifying and testing indirect effects in simple mediation models when the constituent paths are nonlinear. *Multivariate Behavioral Research*, 45(4), 627-660.

Jarque, C.M., & Bera, A.K. (1980). Efficient tests for normality, homoscedasticity and serial independence of regression residuals. *Economics Letters*, 6(3), 255-259.

Kaiser, H.F. (1958). The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23(3), 187-200.

Kaiser, H.M. (2010). *Mathematical programming for agricultural, environmental, and resource economics*. Hoboken, NJ: Wiley.

Keil, M., Tan, B.C., Wei, K.-K., Saarinen, T., Tuunainen, V., & Wassenaar, A. (2000). A cross-cultural study on escalation of commitment behavior in software projects. *MIS Quarterly*, 24(2), 299–325.

Ketkar, S., Kock, N., Parente, R., & Verville, J. (2012). The impact of individualism on buyer-supplier relationship norms, trust and market performance: An analysis of data from Brazil and the U.S.A. *International Business Review*, 21(5), 782–793.

Kim, M.J., Park, C.G., Kim, M., Lee, H., Ahn, Y.-H., Kim, E., Yun, S.-N., & Lee, K.-J. (2012). Quality of nursing doctoral education in Korea: Towards policy development. *Journal of Advanced Nursing*, 68(7), 1494-1503.

Klaassen, C.A., Mokveld, P.J., & Es, B.V. (2000). Squared skewness minus kurtosis bounded by 186/125 for unimodal distributions. *Statistics & probability letters*, 50(2), 131-135.

Kline, R.B. (1998). *Principles and practice of structural equation modeling*. New York, NY: The Guilford Press.

Kock, N. (2011). A mathematical analysis of the evolution of human mate choice traits: Implications for evolutionary psychologists. *Journal of Evolutionary Psychology*, 9(3), 219-247.

Kock, N. (2011b), Using WarpPLS in e-collaboration studies: Mediating effects, control and second order variables, and algorithm choices. *International Journal of e-Collaboration*, 7(3), 1-13.

Kock, N. (2013). *Advanced mediating effects tests, multi-group analyses, and measurement model assessments in PLS-based SEM*. Laredo, TX: ScriptWarp Systems.

Kock, N. (2013b). Using WarpPLS in e-collaboration studies: What if I have only one group and one condition? *International Journal of e-Collaboration*, 9(3), 1-12.

Kock, N. (2014). *A note on how to conduct a factor-based PLS-SEM analysis*. Laredo, TX: ScriptWarp Systems.

Kock, N. (2014b). *Stable P value calculation methods in PLS-SEM*. Laredo, TX: ScriptWarp Systems.

Kock, N. (2014c). *Single missing data imputation in PLS-SEM*. Laredo, TX: ScriptWarp Systems.

Kock, N. (2014d). *One-tailed or two-tailed P values in PLS-SEM?* Laredo, TX: ScriptWarp Systems.

Kock, N., & Gaskins, L. (2014). The mediating role of voice and accountability in the relationship between Internet diffusion and government corruption in Latin America and Sub-Saharan Africa. *Information Technology for Development*, 20(1), 23-43.

Kock, N., & Lynn, G.S. (2012). Lateral collinearity and misleading results in variance-based SEM: An illustration and recommendations. *Journal of the Association for Information Systems*, 13(7), 546-580.

Kock, N., & Verville, J. (2012). Exploring free questionnaire data with anchor variables: An illustration based on a study of IT in healthcare. *International Journal of Healthcare Information Systems and Informatics*, 7(1), 46-63.

Lee, D.T., & Schachter, B.J. (1980). Two algorithms for constructing a Delaunay triangulation. *International Journal of Computer & Information Sciences*, 9(3), 219-242.

Lindell, M., & Whitney, D. (2001). Accounting for common method variance in cross-sectional research designs. *Journal of Applied Psychology*, 86(1), 114-121.

Lohmöller, J.-B. (1989). *Latent variable path modeling with partial least squares*. Heidelberg, Germany: Physica-Verlag.

MacKinnon, D.P., Krull, J.L., & Lockwood, C.M. (2000). Equivalence of the mediation, confounding and suppression effect. *Prevention Science*, 1(4), 173-181.

Miller, R.B., & Wichern, D.W. (1977). *Intermediate business statistics: Analysis of variance, regression and time series*. New York, NY: Holt, Rihehart and Winston.

Mueller, R.O. (1996). *Basic principles of structural equation modeling*. New York, NY: Springer.

Nevitt, J., & Hancock, G.R. (2001). Performance of bootstrapping approaches to model test statistics and parameter standard error estimation in structural equation modeling. *Structural Equation Modeling*, 8(3), 353-377.

Newman, D.A. (2014). Missing data: Five practical guidelines. *Organizational Research Methods*, 17(4), 372-411.

Nunnally, J.C. (1978). *Psychometric theory.* New York, NY: McGraw Hill.

Nunnally, J.C., & Bernstein, I.H. (1994). *Psychometric theory*. New York, NY: McGraw-Hill.

Ogasawara, H. (1999). Standard errors for the direct oblimin solution with Kaiser's normalization. *Japanese Journal of Psychology*, 70(4), 333-338.

Pearl, J. (2009). *Causality: Models, reasoning, and inference*. Cambridge, England: Cambridge University Press.

Peterson, R.A., & Yeolib, K. (2013). On the relationship between coefficient alpha and composite reliability. *Journal of Applied Psychology*, 98(1), 194-198.

Petter, S., Straub, D., & Rai, A. (2007). Specifying formative constructs in information systems research. *MIS Quarterly*, 31(4), 623-656.

Preacher, K.J., & Hayes, A.F. (2004). SPSS and SAS procedures for estimating indirect effects in simple mediation models. *Behavior Research Methods, Instruments, & Computers*, 36(4), 717-731.

Rencher, A.C. (1998). *Multivariate statistical inference and applications*. New York, NY: John Wiley & Sons.

Rigdon, E.E. (2012). Rethinking partial least squares path modeling: In praise of simple methods. *Long Range Planning*, 45(5), 341-358.

Robert, C.P., & Casella, G. (2010). *Monte Carlo statistical methods*. New York, NY: Springer.

Rohatgi, V.K., & Székely, G.J. (1989). Sharp inequalities between skewness and kurtosis. *Statistics & Probability Letters*, 8(4), 297-299.

Rönkkö, M., & Evermann, J. (2013). A critical examination of common beliefs about partial least squares path modeling. *Organizational Research Methods*, 16(3), 425-448.

Rosenthal, R., & Rosnow, R.L. (1991). *Essentials of behavioral research: Methods and data analysis*. Boston, MA: McGraw Hill.

Sarstedt, M., & Ringle, C.M. (2010). Treating unobserved heterogeneity in PLS path modeling: A comparison of FIMIX-PLS with different data analysis strategies. *Journal of Applied Statistics*, 37(8), 1299-1318.

Schumacker, R.E., & Lomax, R.G. (2004). *A beginner's guide to structural equation modeling*. Mahwah, NJ: Lawrence Erlbaum.

Sijtsma, K. (2009). On the use, the misuse, and the very limited usefulness of Cronbach's alpha. *Psychometrika*, 74(1), 107-120.

Spirtes, P., Glymour, C., & Scheines, R. (1993). *Causality, prediction and search*. Berlin, Germany: Springer-Verlag.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, Series B*, 36(1), 111–147.

Temme, D., Kreis, H., & Hildebrandt, L. (2006). *PLS path modeling – A software review*. Berlin, Germany: Institute of Marketing, Humboldt University Berlin.

Tenenhaus, M., Vinzi, V.E., Chatelin, Y.-M., & Lauro, C. (2005). PLS path modeling. *Computational Statistics & Data Analysis*, 48(1), 159-205.

Theil, H. (1958). *Economic forecasts and policy*. Amsterdam, Netherlands: North-Holland.

Wagner, C.H. (1982). Simpson's paradox in real life. *The American Statistician*, 36(1), 46–48.

Wetzels, M., Odekerken-Schroder, G., & van Oppen, C. (2009). Using PLS path modeling for assessing hierarchical construct models: Guidelines and empirical illustration. *MIS Quarterly*, 33(1), 177-196.

Wold, H. (1980). Model construction and evaluation when theoretical knowledge is scarce. In J. Kmenta and J. B. Ramsey (Eds.), *Evaluation of econometric models* (pp. 47-74). Waltham, MA: Academic Press.

Wold, S., Trygg, J., Berglund, A., & Antti, H. (2001). Some recent developments in PLS modeling. *Chemometrics and Intelligent Laboratory Systems*, 58(2), 131-150.

Wooldridge, J.M. (1991). A note on computing r-squared and adjusted r-squared for trending and seasonal data. *Economics Letters*, 36(1), 49-54.